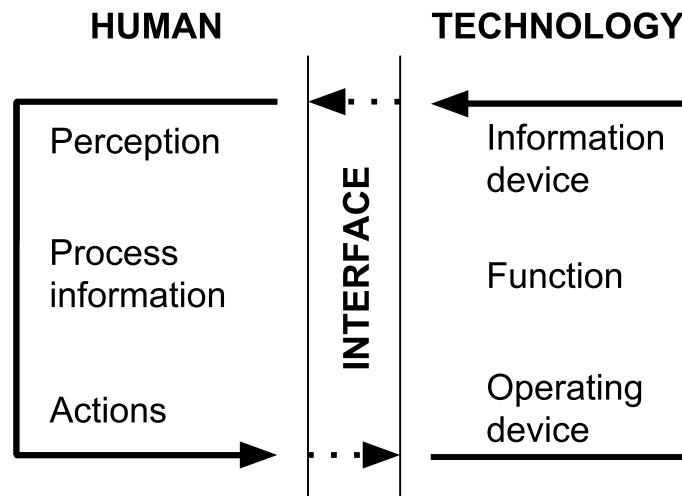




CHALMERS
UNIVERSITY OF TECHNOLOGY



Evaluating the Usability of a Generic User Interface for Discrete Event Simulation

Master's thesis in Industrial and Materials Science

EMANUEL LUNDÉN

ARTHUR WESSMAN

MASTER'S THESIS 2018:IMSX30

Evaluating the Usability of a Generic User Interface for Discrete Event Simulation

EMANUEL LUNDÉN
ARTHUR WESSMAN



CHALMERS
UNIVERSITY OF TECHNOLOGY

Department of Industrial and Materials Science
CHALMERS UNIVERSITY OF TECHNOLOGY
Gothenburg, Sweden 2018

Evaluating the Usability of a Generic User Interface for Discrete Event Simulation
EMANUEL LUNDÉN
ARTHUR WESSMAN

© EMANUEL LUNDÉN, 2018.

© ARTHUR WESSMAN, 2018.

Supervisors:

Emma Strömblad, Logistics Engineering, Volvo Trucks.

Leo Adelbäck, Production and Logistics, ÅF.

Camilla Lundgren, Department of Industrial and Materials Science, Chalmers.

Managers:

Magnus Höglin, Logistics Engineering, Volvo Trucks.

Pär Ström, Production and Logistics, ÅF.

Examiner:

Anders Skoogh, Director, Department of Industrial and Materials Science, Chalmers.

Master's thesis 2018:IMSX30

Department of Industrial and Materials Science

Chalmers University of Technology

SE-412 96 Gothenburg

Telephone +46 31 772 1000

The Authors grant to Chalmers University of Technology and University of Gothenburg the non-exclusive right to publish the Work electronically and in a non-commercial purpose make it accessible on the Internet. The Authors warrant that they are the authors to the Work, and warrant that the Work does not contain text, pictures or other material that violates copyright law.

The Authors shall, when transferring the rights of the Work to a third party (for example a publisher or a company), acknowledge the third party about this agreement. If the Authors have signed a copyright agreement with a third party regarding the Work, the Authors warrant hereby that they have obtained any necessary permission from this third party to let Chalmers University of Technology and University of Gothenburg store the Work electronically and make it accessible on the Internet.

Cover: Image of a human-technology system where human and technology interact in an environment to solve different tasks (Osvalder and Ulfvengren 2015). Information, matter or energy (or any combination of these) are continuously exchanged in this cyclic interaction between the human and the technology under influence from the

environment. The operator (human) has control of the system and its technology. Status and information are displayed via the interface (for example screens). The human takes in the information, processes, and interprets it, which results in an action. The action is done through different types of regulators, upon which the technology part processes it and shows a new status.

Typeset in L^AT_EX
Gothenburg, Sweden 2018

Abstract

The purpose of this thesis is to make the use of a generic Discrete Event Simulation (DES) User-Interface (UI) for AutoMod (Adelbäck and Malmgren 2015) more effective and improve methodology by evaluating its usability. It is carried out for the process of developing a simulation model, by creating a simulation model for one of the case companies. The simulation model is aimed to answer the maximum capacities of the system, and investigate future-state ideas. Focus lies on the evaluation, but the simulation results are also presented. The evaluated tasks are programming the input-variable (from the UI to the model) and output-variable process (from the model to the UI). The evaluation was done to increase programming effectivity and thus decrease programming-time and therefore decrease costs.

Methodology from Banks (1998a) was used to create the simulation model and involves creating a model specification, model building, data collection, verification, validation, planning experiments, carrying out experiments and analyzing these. The planned experiments were re-routing entities via a new and shorter route, and decreasing the cycle-time (C/T) of the believed bottleneck. The bottleneck analysis was done through looking at utilization factors of the resources.

Osvalder, Rose, and Karlsson (2015b) is used to conduct an interaction analysis by analytically evaluating the UI with Predictive Human Error Analysis (PHEA). It was done through defining the evaluation, describing the human-technology system, generating errors, PHEA, and finally discussing the results.

The finished model resulted in the maximum capacities for the current-state. The re-routing of entities in the model had little to no effect for these capacities. The believed bottleneck was proven the real bottleneck in the model, and decreasing its C/T had a significant effect on the maximum capacities.

As for the evaluation, relevant errors from creating the simulation model and their reason, consequence, discovery and correction was revealed using PHEA.

It was concluded that decreasing the C/T of the believed and true bottleneck of the system is recommended, while being mindful of new bottlenecks being created upstream.

Furthermore, the recommendations from the results were to create a programmer's manual for the UI, help the user to match the UI and the AutoMod environments to each other, improve error messages given by the UI, and properly describing the human-technology system for the intended users.

Apart from providing one of the case companies with a simulation model which is a decision-making tool, this thesis have continued the development of the generic DES UI by Adelbäck and Malmgren (2015). This is the contribution to current research.

Keywords: Discrete Event Simulation, DES, AutoMod, Simulation, User Interface, UI, Production, Usability, Evaluation, Cognitive Walkthrough, CW, Predictable Human Error Analysis, PHEA, human-technology system.

Acknowledgements

Volvo Trucks (Tuve) supervisor Emma Strömblad, manager Magnus Höglin, W03 warehouse expert Kjell Dahl. The rest at Volvo Logistics Department. All the workers at the W03 warehouse.

Chalmers supervisor Camilla Lundgren and examiner Anders Skoogh.

ÅF supervisor Leo Adelbäck and manager Pär Ström.

Opposition group Pontus Helmersson and Johanna Hesselund.

Thanks to all of you for making this master's thesis possible!

Emanuel Lundén and Arthur Wessman, June 2018.

Contents

1	Introduction	1
1.1	Background	1
1.2	Purpose	2
1.3	Scope and limitations	3
1.3.1	Time limit	4
1.3.2	Cost limit	4
1.3.3	Constant triangle	4
1.3.4	Stakeholder analysis	5
1.3.5	Business objectives	5
1.3.6	Project objectives	5
1.4	Research questions	6
2	Theory	7
2.1	Theory of Constraints	7
2.2	Exploratory learning	8
2.3	Human–technology system	8
2.4	User interface	9
2.4.1	Input	10
2.4.2	Output	11
2.4.3	Results	11
2.4.4	Simulation	11
2.4.5	MultiSim	12
3	Methodology	13
3.1	Simulation project	13
3.1.1	Softwares	14
3.1.2	Design Phase	14
3.1.3	Analyze Phase	19
3.2	Bottleneck analysis methods	20
3.2.1	Utilization factor	20
3.3	Evaluation	21
3.3.1	User profile	21

3.3.2	Hierarchical Task Analysis	22
3.3.3	Cognitive Walkthrough	23
3.3.4	Predictive Human Error Analysis	24
3.4	Interaction analysis by analytically evaluating the UI with CW and PHEA	25
3.4.1	Definition of the evaluation	25
3.4.2	Describing the human-technology system	26
3.4.3	Predictive Human Error Analysis	26
3.4.4	Conclusion and follow-up	27
4	Results	28
4.1	Simulation model	28
4.2	UI evaluation through PHEA	30
4.2.1	Input process	31
4.2.2	Output process	32
5	Discussion	34
5.1	Method	34
5.2	Result	35
5.2.1	Simulation model	35
5.2.2	Evaluation	36
5.3	Sustainability	37
5.4	Contribution to current research	39
6	Conclusion	40
	Bibliography	42
	Appendices	45
A	Stakeholder analysis	i
B	CW and PHEA form	ii
C	Evaluation method	iii
D	HTA for the UI input process	iv
E	HTA for the UI output process	v
F	UI - Main page	vi
G	UI - Sheet handler	vii
H	UI - Control sheet	viii
I	UI - MultiSim	ix
J	Graphical model of the system simulated	x
K	Simulation results	xi

Abbreviations

DES - Discrete Event Simulation

C/T - Cycle time.

MHS - Manual handling station.

TTR - Time-to-repair.

TBF - Time-between-failure.

MTM - Methods-time measurement.

UI - User interface.

HTA - Hierarchical Task Analysis.

CW - Cognitive Walkthrough.

PHEA - Predictive Human Error Analysis.

1

Introduction

1.1 Background

Developed and popularized from the end of the 1950s, Discrete Event Simulation (DES) has grown into the most used way of researching operations for a wide range of industries according to Hollocks (2006). This is more or less the case in Sweden as explained by Eriksson (2005), but for example only used regularly by a couple of companies, often overshadowed by other methods of decision support, and still emerging in Swedish industry even though existing for quiet some time. Despite this, it is one of the three most popular tools in management science.

DES is taught in Sweden on for example Chalmers University of Technology as seen in Chalmers (2016) at master level, and there is a need for DES models to be created and used in master theses to this day as shown in Mansour and Moreira (2017), Goodwin and Pantzar (2017), as-well as Dawson and Lindahl (2017).

This master's thesis is conducted at Volvo Trucks in Tuve, Göteborg. Volvo Trucks is a world-wide truck manufacturer competing on a global market that expresses a general need of continuously improving, much like any other company today. Their production department works on a number of tasks to meet these general goals, one of which is mapping in- and outbound capacities for different parts of the facility in order to make improvements in the correct areas. A part of the production where these capacities are not completely investigated is an automated small-box warehouse called W03 consisting of conveyors, cranes, storage, a pallet breakdown cell and machines filling transportable racks.

The exploration of the maximum capacities of the warehouse is intended to be done through a DES model, which will also act as a foundation for finding improvements and future investments. Specifically, they have interest in finding out the effect of two improvements. A shorter and separate path for reject boxes, and a fourth crane with

storage.

ÅF is using a user interface (UI) described in Adelbäck and Malmgren (2015) and it is used for this project. It is used regularly in AutoMod production DES projects conducted by ÅF. The UI allows the customer to run the model, change variables and interpret output data from the same interface with the cheaper run-time key instead of having to use a developer key to change fixed variables in the AutoMod environment. In addition to using it and creating a simulation model, the usability of the UI is evaluated for ÅF. This to contribute to the improvement of their methods to enable effective use of them, but also because the UI has never been evaluated for new users such as the authors.

Furthermore, it is used consistently by the simulation programmers at ÅF and by their customers. As seen in Montgomery (2011), the impact of the UI usability on simulation quality is very large, and motivates usability of the UI for the customer using it. This does not motivate improvement of the usability of programming the UI. However, programming time is related to the cost of the project, and thus desirable to decrease by for example making the programming of the UI more effective. Finally, the usability was not evaluated by Adelbäck and Malmgren (2015) apart from user interviews, further motivating the conducting of an usability evaluation.

Volvo Trucks is the end-customer of the production DES model, but it is supervised by the consultancy company ÅF. The evaluating of the UI is aimed towards ÅF and Chalmers.

The project aims to contribute to current research by evaluating a newly created UI (2015) for production DES and AutoMod, and with this also continue the UI development process. This can be of value for other projects that develops DES UIs, or other projects dealing with usability and UIs where human-technology system methodology is implemented.

1.2 Purpose

The main purpose of the project is to improve methodology and effective use of a generic DES UI by evaluating the usability for its implementation throughout the development of a simulation model for a production system at one of the case companies.

The simulation model will be used to find the maximum capacities of the studied system, the bottleneck, and the effect of two future-state scenarios.

1.3 Scope and limitations

There exist limitations on what is included in the model. These are best seen in the model specification (which is not included in the report due to confidentiality), but generally consist of the warehouse itself, its machines and human processes within its boundaries.

Only one iteration with bottleneck analysis and generating improvements will be done. The system will not be balanced, and there will be no attempts to increase the overall capacity (only simulating the future-state improvements). There will only be two simulated future-state scenarios, namely a shorter and separate reject path, as-well as the effects of an additional crane with storage.

Modeling humans is simplified to using MTM times produced by Volvo Trucks, and no attempt will be done to improve these times or processes.

Downtime is included in the model and is limited to what is seen in the model specification, which is not displayed here due to confidentiality reasons. Focus lies not on making breakdowns statistically valid, but to make the future state trustworthy and prevent just improving the process based on which resource has the longest cycle time. Maintenance will thus not be improved, since breakdowns are of lesser importance for the company and the system.

In the end, it is the responsibility of Volvo Trucks to provide all the available and relevant data to the project. The project will attempt to collect the remaining data, but is not guaranteed or be of the same quality.

Due to reasons such as time-limitations and relevancy, the evaluation will include the processes of programming the input, as-well as the output. For time and effectivity reasons, the users considered and evaluated in the evaluations are the authors exclusively.

The current-state and maximum capacities will only contain one scenario that focuses on maximum input and output simultaneously, and all other variables set to what is considered normal production.

The model will be able to display all maximum capacities for different types of scenarios. Because of limitations in the UI, the report will display maximum capacities for boxes leaving cranes, boxes leaving racks, and boxes coming in to the system.

1.3.1 Time limit

The project is aimed to be 30 credits, which is equivalent to one semester (20 weeks). The project started 15-01-2018 and ended 15-06-2018.

1.3.2 Cost limit

There are no costs for the project as equipment and software are provided by the stakeholders.

1.3.3 Constant triangle

The *constant triangle* in Figure 1.1 shows, if conflict arise, that focus will lie on reaching project objectives (quality), followed by time and then resources. This does not imply that there are infinite resources or time, but means that achieving qualitative results within a smaller scope. Compare this to for example reaching acceptable quality on a larger scope. Additionally, if there is a conflict when for example choosing between two alternatives or solutions, the most important part is quality for the customer. The project will however have to fit inside the given time-span of the master's thesis project.

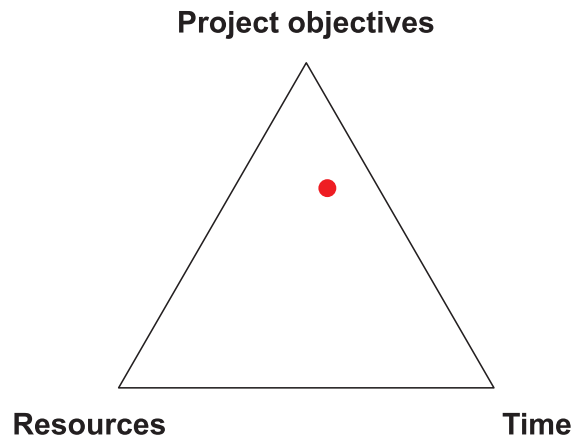


Figure 1.1: The constraint triangle showing that focus lies on the project objectives (quality).

1.3.4 Stakeholder analysis

In order to understand the stakeholders, a stakeholder analysis is conducted. It involves identifying needs and expectations, grouping them into result or realization, as-well as primary or secondary. The result is displayed in Appendix A. The main stakeholder is Volvo Trucks, and the stakeholder customer is ÅF. Chalmers is a stakeholder interested in the methodology, realization, and result, but also the research value of the project.

1.3.5 Business objectives

Providing Volvo Trucks with maximum capacity for different parts of the system is the main business objective, as-well as identifying bottlenecks and provide improvement suggestions. The reason for this is that they are unsure where issues exist in the process, and the desirability for testing how changes to the system affect the flow and maximum capacity.

A simulation model will be created in order to fulfill the above objective. This will be measurable by an increase in throughput, a better flow and a more balanced system, compared to the current-state. Long-term utilization of the project's results will lead to an increase in maximum capacity and possibilities for further system improvement. An increase in productivity, effectivity, profitability, etc. and a decrease in waste are some examples of the possible effects that might occur.

ÅF as a consultancy company has interest in making their methods more effective and improving them. The contribution of this report aims to do this by evaluating the UI and follow-up the development of the UI. This will enable improvement work and in the long run may lead to shorter simulation project development times, and in turn decrease in costs.

1.3.6 Project objectives

Listed below, in order of importance, are the project objectives.

- **Simulation model**

Producing a simulation model of the current-state warehouse at the case company is the main objective.

- * **Validation**

- To prove its accuracy, the created simulation model have to be compared and validated towards existing data.

- * **Documentation on how the simulation model is built and manual**

- This is critical for for the case company to be able to use and understand the model.

- * **Data collection**

- With faulty or low quality data, the quality of the model can be insufficient and the model itself can be questionable.

- **Analyzing**

Simulation runs, affecting factors and synergy, experiment design, statistical groundwork, improvement suggestions, future state, are important results for the customer.

- **Report**

For the stakeholders to be able to see what has been done.

- **Improving simulation project methodology**

The methodology behind working with simulation projects and how and if there are any improvements to be made to the current methodology, is something of interest to ÅF.

1.4 Research questions

Given the earlier sections, the research questions are as follows.

1. What is the maximum capacity of the warehouse?
2. How can the maximum capacity be improved?
3. What are the errors that occur when new users program the UI?
4. How might these errors be eliminated?

2

Theory

In this chapter, the theoretical foundation for the project is explained to be able follow the methodology in the next chapter. Section 2.1-?? is needed to fulfill the bottleneck analysis objective described in Section 1.3.5 and Section 1.3.6. Section 2.2 and following are all theory for the evaluation.

2.1 Theory of Constraints

As previously described, the simulation model is to be improved. The underlying theory and assumptions for continuously improving a production system by reducing losses as seen in Goldratt and Cox (2004) are described in the following steps.

1. Identify the constraint (bottleneck), the weak links in the chain that are worsening the fulfillment of goals and KPIs.
2. Decide how to exploit the constraint by doing what is possible to improve the constraint here and now. Done without elevating, which means without investments.
3. Subordinate other resources to the constraint. Make sure that the flow up- and downstream of the constraint allows for it to never starve and never be blocked.
4. Elevate the constraint. Finally taking to bigger investments of different kinds, often capital ones.
5. If in any of the previous steps a constraint is broken (eliminated), go back to step 1.

Revealed in Lima, Chwif, and Barreto (2008) and Roser and Nakano (2015) are that the most recommended methodology for detecting bottlenecks of different kinds (including shifting ones) in Step 1. is the *active period method* described in Roser, Nakano, and Tanaka (2002) for this type of project where there are little to no product mixes, and fluctuations are relatively low. The other bottleneck detection method recommended

in Lima, Chwif, and Barreto (2008) for this type of project is looking at the *utilization factor*, where the resource with the highest utilization factor is the bottleneck.

In this report, Step 1 and 4 is mainly considered from this theory, based on the demands of the case-company. This is because only a couple of scenarios are explored, and focus lies on finding the bottleneck, not improving the bottleneck.

2.2 Exploratory learning

There are underlying philosophies and assumptions as to why usability evaluation is needed. These are interesting to bring up as they explain the mindset that evaluators have, and the prerequisites before even starting working with evaluation. According to diSessa et al. (1995), the learning philosophies as follows are what defines *exploratory learning*.

- Learning can to a large extent be under control by the learners, that they are able intellectual persons and should be treated as such.
- Multidimensional and rich are two characteristics of knowledge. There are many paths to enlightenment.
- Learners are diverse. Their intellectual style should be respected, regardless of number of achievements.
- The possibility of learning feeling easy and natural can often or always exist in a context of proper physical and cultural design.

2.3 Human–technology system

Describing the system is necessary when conducting an evaluation of a UI, which is one thing that this report aims to do as previously described. A system where human and technology interacts is called a *human-technology system* as seen in Osvalder and Ulfvengren (2015). Here technology means anything from simple tools to aircraft technology in a cockpit. Everything from simple individual components like a coffee-brewing machine, to technically complex systems in a nuclear plant. A detailed description and illustration of how a human-technology system works is shown in Figure 2.1.

Osvalder and Ulfvengren (2015) continues that *usability* is necessary in order for the human-technology system to fulfill its purpose, and is split up into two parts, namely the correct and working functionality, and usefulness which addresses the simplicity and safety for the targeted group.

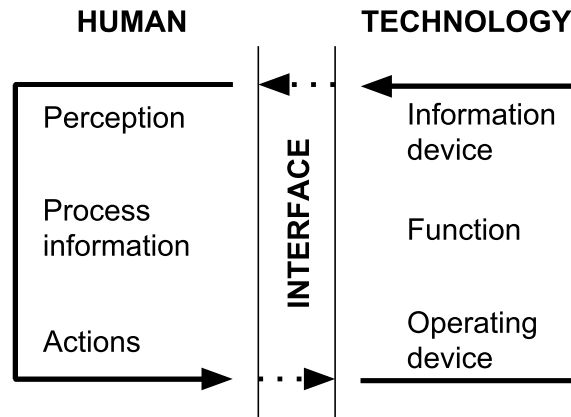


Figure 2.1: A human-technology system where human and technology interact in an environment to solve different tasks (Osvalder and Ulfvengren 2015). Information, matter or energy (or any combination of these) are continuously exchanged in this cyclic interaction between the human and the technology under influence from the environment. The operator (human) has control of the system and its technology. Status and information are displayed via the interface (for example screens). The human takes in the information, processes, and interprets it, which results in an action. The action is done through different types of regulators, upon which the technology part processes it and shows a new status.

Finally, Osvalder and Ulfvengren (2015) explains that maximizing the human’s capability of handling the system while minimizing the effects of the human’s limitations are the purpose of the human-technology system as a subject. Central knowledge involves the human’s capability of solving the mental and physical tasks required of the system, while being affected by physical and social environmental factors. Technical psychology is included in human-technology systems as further explained in Danielsson (2001), and human-technology systems are generally defined by cognitive prerequisites, abilities and limitations when it comes to processing information and making decisions.

2.4 User interface

The user interface that is used for changing input variables and interpreting output data and results made by Adelbäck and Malmgren (2015) can be seen in Appendix F. The main page of the UI consists of four buttons; *Start*, *Results*, *Input* and *Output*. Unique for the Start tab is that there is small window prompting the user that it has to have the correct system separator chosen for the UI to operate as intended.

When a button is clicked the user is brought in to the UI for that particular category.

Each category got a different color scheme as well as individual excel sheets and can be seen in Figure 2.2. It is through these sheets the different variables and data can be accessed and stored.

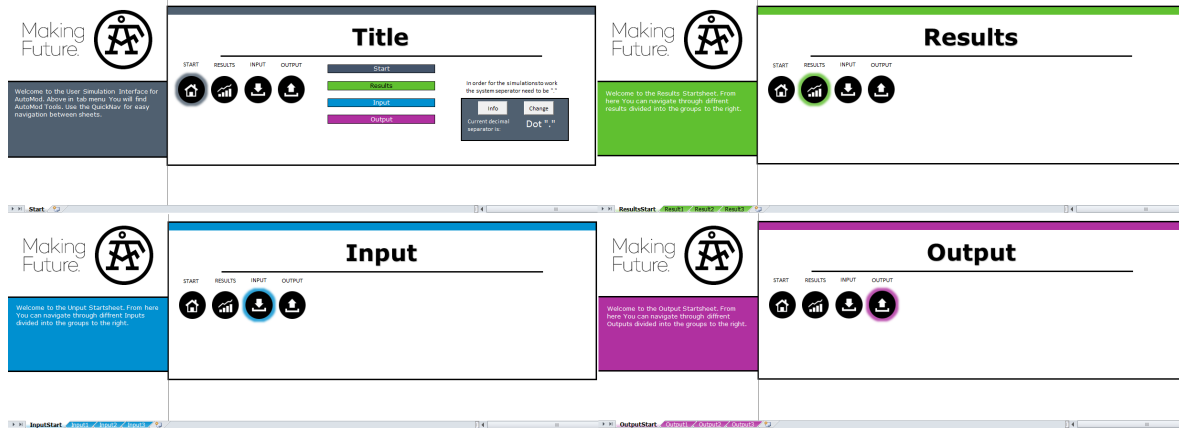


Figure 2.2: The four different UI menus and categories in the UI.

2.4.1 Input

Below follows a brief description of the process of adding Input variables to the UI as well as connecting them in the AutoMod code. For a detailed description of the process please refer to the HTA in Appendix D.

The first step in the input creation process is to go to the menu and click *QuickNav* then click *UI Options*. This will reveal three excel-tabs named *ControlSheet*, *SheetHandler* and *MultiSimSetup*. The second step is to go into the *SheetHandler* tab and click *New sheet* on the Input column (can be seen in Appendix G). When the *New sheet* button is pressed an empty excel sheet is created. Now it is up to the user to name this sheet with a name that fits the variables that should be stored under that sheet. After the new sheet has been added to the Input column in the *SheetHandler* tab it is time to enter data. Now data and descriptions from the data collection phase is added to the sheet.

The next step is to make sure that the UI extract the correct values from the sheet. In order to do this the user has to go to the *ControlSheet*, see Appendix H. Here the user has to choose their current sheet from a dropdown menu under *Sheet Name* then choose an appropriate name for the text-file that the UI creates upon simulation. Finally the cell numbers from which cells that should be printed to the text-file has to be entered. Here it is also possible to select writing method dependent on whether the user wants to write all cells or only cells that are unprotected.

When all this is done what is left is to make AutoMod read the data from the text-files. In order to do this the user has to go to the AutoMod code and program AutoMod to read the data from each text-file. Variables have to be created and matched with the correct line and value in the text-file.

2.4.2 Output

The Output process is similar to the Input process and can be seen in the HTA in Appendix E. It starts of the same way as the Input creation where a new sheet is created in the *SheetHandler* sheet. The sheet is given a name and is added to navigation according to the HTA. Next step is to format the sheet to fit the data the model will output. This involves creating columns and rows to correspond to specific output data as well as making enough of them to fit the corresponding *snap* interval and length of the simulation. Here, snap interval means the set interval when to capture the output data collected for the duration of the snap. When the sheet is created the user has to go to the AutoMod code and code how the output should be made dependent on how the sheet format was created as well as naming the text-file the output-data should be written to. When this is done the final step is to make sure that the UI can read the data from the text-file. This is done in the same way as for the Input process. Under the *ControlSheet* the sheet name and the text-file has to be entered as well as what cells the data should be written to, see Appendix H.

2.4.3 Results

The results sheet have no settings or controls directly connected to it and its sole purpose is to be used as a place to present and format the output data in a more compact and easily presented view.

2.4.4 Simulation

After all the input and output data and formatting is done there is one step left before the UI can run the simulation. The final step is to add the name and path of the folder containing the simulation model. The user has to go to the *ControlSheet* and enter the model name, path, input path as well as output path. Now the UI can access the files needed for running the simulation and in order to start the simulation the user has to go to the *AutoMod Tools* tab in the upper parts of excel, see Figure 2.3. When the Simulation button is pressed the UI will create all the text-files based on the data

in each Input sheet and open the AutoMod simulation. When the simulation is done and closed, the UI will read the output data from the text-files and add it to their corresponding sheet based on the settings entered in the *ControlSheet*.

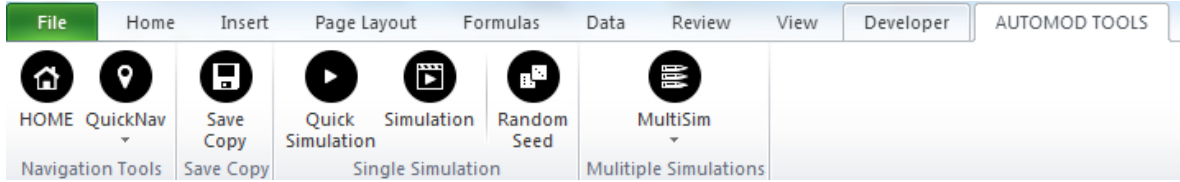


Figure 2.3: Display of the additional Excel toolbar that the UI provides called *AutoMod Tools*.

2.4.5 MultiSim

The *MultiSim* feature of the UI is used for analyzing how changes between variables and runs affect certain outputs. The user is presented with a couple of different choices that can be seen in Appendix I. There are two different kinds of analysis type, *Vary factors* and *Multiple runs*.

Vary factors is dependent on the user selecting factors from the input data. The user have to select the cell from any of the input sheets and specify the value that should be tested each run. Other than that, the user can also change how many repetitions should be run for each factor and if the simulation should be run with a random seed as well as how many factors should be used and how many responses the run should result in. The random seed setting is used for making the simulations run differently, or similarly if the variables and seed are all kept the same.

The analysis type *Multiple run* the UI will change and remove the factor cells as well as cross out random seed and number of factors. This is because during *Multiple runs* no input data is changed and the only changeable variables are number of repetitions and the response cells. For a more detailed explanation of the *MultiSim* feature please refer to the report by Adelbäck and Malmgren (2015).

3

Methodology

In this chapter the course of action and chosen methods throughout the project is presented. As explained in the introduction, the overall aim is to create a simulation model and evaluate the general UI by Adelbäck and Malmgren (2015). A methodology foundation is provided and required by ÅF described below in Section 3.1, which is mainly based on Banks (1998a), but not limited to that chapter in the book as it refers to other chapters for details. Banks (1998b) was also generally used as guidance. The individual methodology parts of interaction evaluation is described in detail in Section 3.3 and put into context in Section 3.4, which displays how these methods were used in the project. The source for this part is mainly Bohgard et al. (2015). See Figure 3.1 for how the project was carried out. See the following sections for a more detailed description.

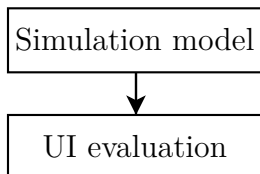


Figure 3.1: The general work-flow for the project, where the simulation model was created and followed by evaluating the programming process of its UI.

3.1 Simulation project

The main methodology used for the simulation project as provided and required by ÅF takes inspiration from Banks (1998a) and other chapters in that book, which is used as the source until otherwise mentioned. It can be concluded and categorized in four main parts. The first two main steps are described in the subsections following the next section about the used software, as the last two steps are formalities for ÅF and not interesting for this report.

1. Design phase.
2. Analyze phase.
3. Documentation phase.
4. Ending phase.

3.1.1 Softwares

In this section, the different relevant software used during the project. *AutoMod* is a discrete event simulation software used for creating the simulation model and simulating (ÅF n.d.).

JMP 13 is a statistical software that can, among many other things according to SAS (n.d.) (which refers to it as *basic data analysis*), display histograms of data sets, fit distributions to them and rank the distributions in the order of which one fits best. It has been used when creating the breakdown statistics.

3.1.2 Design Phase

Creating the current state is the main objective of the design phase. As seen in Figure 3.2, the steps are creating a model specification, followed by model creation, where these two are done in parallel with data collection. Once both are done, one have to verify that the code produced in the model creation follows the model specification, upon which the model needs to be validated to be true to reality. If not valid, the model specification and data collection has to be updated, and the steps that follow are conducted again, until the model is finally validated.

The design phase was conducted in this way, with the difference that this project's model specification and data collection took long time and resulted in them being done in parallel followed up by creating the model. Because the case company consistently provided changes and new material to the model specification, the model was coded to these demands. This means that the process in Figure 3.2 was not followed completely, and essentially, the model was coded before completing the final model specification.

Model specification

Creating the model specification includes conceptualizing the system into a model by the different parts the system consists of, the characteristics of them, and by how they interact, all according to the problem formulation provided by the case company. This

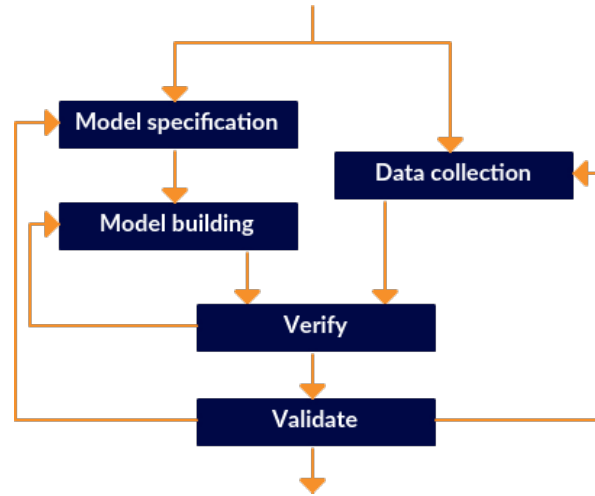


Figure 3.2: Flowchart of the design phase. Model specification and model building are both done parallel to data collection. Following the completion of these are verification, which results in validation if successful, and more model building if failed. The final step is validation and if not passed, updating the model specification and data collection may be necessary. Verification is needed for the model specification to match the model specification, while validation has to do with the model matching reality.

in order to have a solid foundation when starting with the model creation, and for the case company to clearly see what the model have been based upon. As there existed no template of this at ÅF but only some guidelines and recommendations, the model specification were created using the following sections.

1. Function description.
2. Downtime.
3. Input data.
4. Output data.
5. UI changeable variables.
6. Model verification and validation plan.

Here, the function description describes the whole system, its different parts and details on how they interact. Separated from this is the downtime section, which describes how the machines breakdown by distributions for time-to-repair (TTR) and time-between-failure (TBF), how these were created, what the sources for the breakdowns are, what assumptions and simplifications were made, etc.

The following two sections, input and output data, briefly describe what the model takes in and what the result is. In this case, the input data can be how often pallets arrive to the first station, when errors occur for the boxes at the scanner, how often boxes are ordered from the storage, what destination they have, and so on. The output data

describes what should be the result of the model runs, and can be for example a graphical representation of the states and their proportions for each machine, throughput for boxes leaving system, and average time for a box to get there.

The UI changeable variables' section list for example what variables are changeable in the UI, what format they have and how the output variables are going to be presented. Compared to the previous two sections, input and output data, these variables are only changeable for the purposes of analysis and the future state, which is why they have their own section. For example, it can be different cycle times which should not be altered for the current state unless erroneous due to aspects such as them changing over time or being faulty in the model specification.

Finally, here is also the preliminary verification and validation plan presented to be observed and discussed with the case company stakeholders. It can include what ways to verify and validate, as-well as how they work.

Model building

Following the model specification is the actual model creation. Without going into too many obvious underlying assumptions of how to write this or other types of code, Banks (1998b) entail that some recommendations are to focus on the problem, start simple with pseudo code, test general functionality before going into more complex and detailed parts, and finally review efforts and effectively realign with project objectives.

Data collection

In order for the model specification and in turn the model to be realistic, one must collect data or information of different types. Methods of collecting can be observing the system, clocking times, asking questions, reading manuals, or making assumptions when data is not found. Data or information collected can be how the system operates on a wide scale to a detailed level, cycle times, buffer capacities, different kinds of speeds, breakdown statistics and so on.

Interesting times of mechanical and automated elements were clocked by the authors, with enough samples (usually 5-10) depending on the variance experienced. However, because of reasons such as data being erroneous, and data changing over time, the studied time interval for the breakdown statistics was fixed to a little over a month in order to create them. The breakdown statistics have been affected negatively from this as the considered time period is relatively short, which effects the statistical validity of them. It is however of lesser significance as the main focus is the current state and

maximum capacities. This is further motivated by the resulting breakdown distributions as they display short and infrequent downtimes. In addition, the breakdown data were modified to remove occurrences of very short TTR and TBF, as they are erroneous and a result of for example sensors with bad contacts or errors generally being replicated many times with a couple of seconds TTR and TBF. Breakdowns are however included and is important in order to create a realistic future state. All of the times, speeds and breakdown distributions were left changeable in the UI and thus allowing the case company to improve them after the completion of the model.

Verification

The first step towards getting the model finalized is called verification. It is done by comparing the model specification to the computerized model and making sure that they correspond to each other. Errors found in the computerized model have to be changed and updated. Following the verification process is the validation, which makes sure that the model specification and computerized model corresponds to reality. Explained in Sargent (2013), the methods for verifying the computerized model are called *trace* and *structural walkthrough*. Trace is when an entity in the model is followed through the system in order to know if the logic is correct and accurate enough. Structural walkthrough is done by going through the code with a set of peers in order to verify that the code is correct.

The traces done for the computerized model are as follows and involves checking logic for not only the path itself, but for times, resources, activities, and so on.

- A rejected pallet.
- A box to (respectively),
 - cranes,
 - reject,
 - racks,
 - cranes when storage full,
 - Manual handling station (MHS) from cranes,
 - cranes from MHS,
 - racks from MHS.

Validation

After the code itself is verified, Banks (1998a) continues that validation is the next step and makes sure that the model specification is accurate enough in comparison to the real system, and if additional or other types of data collection is needed. Again,

according to Sargent (2013), the validation steps are the following.

1. Validate the conceptual model by testing the assumptions and theories.
 - a. Using statistical methods and mathematical analysis, and review these in turn.
 - b. Having the case company participating in (at least) face validation of the model specification along with the computerized model. Done in every validation iteration.
2. For at least a few experimental condition sets, compare the simulation model and the behaviour of the system (output data) with the real system. This is done in every validation iteration.

Step 1.a. is normally done once before the computerized model creation as explained in Sargent (2013). For this project however, the elements of the model specification containing assumptions, modelling, theories, etc., were reviewed by the case company. This because the project was limited in time, and the authors could only recommend methodology and present the resulting model specification for the case company to take decision upon.

The case-company here settled with providing as good data as possible to the base scenario and assumptions done for it. Face validation was done here as-well as in the validation, and Step 2. above was also conducted.

The recommended validation techniques presented to the company chosen from Sargent (2013) are therefore as follows.

Animation validation: Visually observing the animation in order to validate behaviours. This is done for a short period of simulated time in the simulation model.

Degenerate tests: Validating theoretical bottlenecks by testing with infinite buffers, upon which the bottlenecks should display an increase to infinity as the arrival rate is higher than the service rate.

Event validity: Confirming that occurrences of events in the simulation model are similar to the real system. This is applied to breakdowns, input data, number of boxes going to different destinations, or other entities containing distributions by confirming with future data.

Extreme condition test: Realistic outputs should be obtained in extreme and rare scenarios.

Face validity: Experts on the system reviewing the model to make sure that its behaviour are good enough compared to the real system.

Internal validity: Exploring the stochastic variability from many runs of the model. If too large, the model can be questionable and in turn question the methods or

if the real system itself is worth simulating.

Operational graphics: Displaying text statuses of for example buffer occupancy, storage occupancy, etc. as a tool to aid in validating the model by the authors during running.

Philosophy of science methods: Making sure that the model is logically correct and clearly state assumptions (rationalism), that every assumption and outcome to be empirically validated (empiricism), that the model outcomes are correct without much concern for how it is done (positive economics). Due to time limits and resources, empiricism was probably the least prioritized of the three, this is at least true for cycle times, breakdowns, etc. but does not include for example inputs or outputs.

Predictive validation: Attempting to forecast behaviours using the model, and then compare this with the real system. Can either be done operationally or through experiments.

Turing test: Letting knowledgeable individuals attempt to distinguish the real system from the model outputs.

Out of these recommended validation techniques, the ones the case-company wanted was animation validation, event validity, extreme condition test, face validity, and operational graphics.

3.1.3 Analyze Phase

After the current state have been created in the design phase, the creation of the future state can begin by analyzing the current state in the analyze phase. It consists not only of analysis, but of three parts; namely experiment design, simulation runs of experimental design, and finally analysis. Each of these parts are described in the sections below.

Experimental design

As previously stated, the analyze phase is not entirely analyzing, and importance lies in planning. Specified in this part of the analyze phase is what scenarios to be analyzed, what time-span the runs should have in the simulation, how many runs (replications), how they are initialized, etc.

For this project, there are two scenarios to be simulated apart from analyzing bottle-necks and what measures are possible to balance the system. The first one is evaluating what effect a fourth storage crane and accompanying storage would have, as the cranes

are believed by the case company to be a bottleneck. This will be done by decreasing relevant times by 33% like loading and unloading, and increasing the speed by 33%, effectively decreasing the cycle-time (C/T) by 33%. It was done this way due to time and model-complexity reasons. The second one is rerouting the incoming boxes being rejected to an alternate path. This because the current path more or less goes through the whole system, as-well as taking capacity from both the in- and outgoing conveyors to the cranes.

As it turned out during the course of the project, the active period method described in Section ?? is not implemented for AutoMod, nor has the extraction of data to an external active period algorithm been made. This is why the bottleneck analysis will be carried out by analyzing the utilization factor instead.

Simulation runs and analysis

After the planning of the analyze phase was done in the experimental design, the actual runs and their analysis was conducted. Apart from varying important parameters and otherwise follow the experimental design, it was important to document the runs from the aspects of traceability and reproducibility. It is also necessary to make sure to do enough runs in order to have satisfactory statistical material. The runs was performed through the UI. More planning and runs can be done after this step, if necessary for different reasons.

3.2 Bottleneck analysis methods

In the following sections, the bottleneck methods needed by the theory of constraints shown in Section 2.1 are presented. They are based upon recommendations in Lima, Chwif, and Barreto (2008) for this type of production DES project.

3.2.1 Utilization factor

As described in Section 2.1, methods of finding the bottlenecks are needed, and one of them is described in this section. Described in Lima, Chwif, and Barreto (2008) is one of the recommended bottleneck detection methods when there are few or none product mixes, and fluctuations are relatively low. It is called *utilization factor* and is an easy method with short implementation time as the simulation results from any DES software are the input. The resource that has the highest utilization factor, that has

the highest *active time* of the resources, is the bottleneck. The active time is whenever a resource is not starved, blocked, or otherwise waiting. Downtime is included in the active time. This method was used in order to determine the bottleneck in the system.

3.3 Evaluation

Here, different parts of the method for conducting a usability evaluation through interaction analysis is presented. It involves creating a *user profile*, *hierarchical task analysis diagram* (HTA), *cognitive walkthrough* (CW), and finally a *predictive human error analysis* (PHEA), which are all displayed in their respective sections below.

3.3.1 User profile

When conducting an evaluation, the outcome entirely depends on who the users of the evaluation are and thus there is a need for describing them (Osvalder, Rose, and Karlsson 2015b). As originally explained in Janhager (2005) and brought up in Osvalder, Rose, and Karlsson (2015a), a method for presenting data from studies of the users is called *user profile*. It contains information of a target group's mental, physical and demographic data, and other characteristics like occupation, competence and fulfillment of requirements, etc. that are deemed to have an effect on the evaluation. For the given product, equipment or system that the human interacts with, the related characteristics like knowledge, abilities, and limitations are described. Thus, the aim of the user profile is to map the factors of the interaction.

Osvalder, Rose, and Karlsson (2015a) continues to explain that the types of users involved are important to define because they all have different demands, and can be categorized into four roles, primary (a driller using a drill), secondary (a repairman repairing a drill), side (a person sitting in the next room to the driller), and co-user (person working beside the driller).

Finally, Osvalder, Rose, and Karlsson (2015a) lists the components that the user profile consists of as follows.

1. Background:
 - a. Types of users.
 - b. Gender, age, home district, country.
 - c. Level of education.
 - d. Language skills.
2. Usage:
 - a. Generic knowledge of the product.
 - b. Mental model of the product.
 - c. Experience or training that the users have of the product.
 - d. How frequent the users use this type of product.
3. Influence and responsibility:
 - a. Possibility to affect the choice of product that is used.
 - b. Influence of the usage situation; if the user, someone else or another technology is controlling the situation.
 - c. The responsibility of the product.
4. Emotional relation:
 - a. Ownership.
 - b. Social aspects like if the user wishes to signal something to the environment by using the product.
 - c. Feelings, impressions, opinions or other mental effects that the product can have on the user.
5. Type of interaction:
 - a. The cognitive interaction.
 - b. The physical handling.
 - c. The effect of functional limitation on the possibility to use the product.
6. Activities (what to do with product), goals (goal of using product), and motives (why using this product).

3.3.2 Hierarchical Task Analysis

Naturally, describing the process for carrying out the task is indispensable when carrying out an evaluation. A systematic and structured method for describing how a task is carried out is *hierarchical task analysis* (HTA) described in Stanton (2006). The task is broken down into smaller parts, and it results in an overview of the different functions, how they are built up and their interrelations. Cognitive aspects can be included, but it mostly focuses on physically and observable actions. The method is flexible with an level of detail adjustable to the area of application.

Stanton (2006) continues that an HTA is constructed by having main goal, that is

broken down into sub-goals, where the branches always end with one or more actions. This means that going down the branches answers *how* it is done, while going up answers *why*. Both the goals and actions are written in verb-noun form. Apart from a branch of goals always ending up in an action, a goal cannot have only one sub-goal, a sub-goal can only have one parent goal, and all sub-goals on the same level have the same level of abstraction. The arrows connecting goals can have a text connected to it describing if the goal happens at a specific time, if there is a choice between many goals, or if a certain condition has to be fulfilled. Finally, the HTA is done when the actions reach a level of detail that is not meaningful for upcoming analyses. The structure of an HTA is described in Figure 3.3.

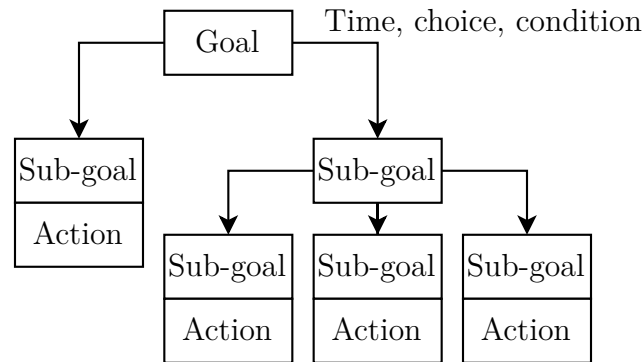


Figure 3.3: Illustration of how a HTA is constructed. Note that a branch of goals always ends up in an action, a goal cannot have only one sub-goal, a sub-goal can only have one parent goal. All of the different arrows connecting goals can have a text connected to it describing if the goal happens at a specific time, if there is a choice between many goals, or if a certain condition has to be fulfilled.

3.3.3 Cognitive Walkthrough

When evaluating, one needs to show the problems that occur when following the given instructions explained by an HTA (Section 3.3.2) to then later be analyzed (in the methodology in next section). Originally described in Lewis and Wharton (1997) and put into context by Osvalder, Rose, and Karlsson (2015b), *cognitive walkthrough* (CW) is today a common method for evaluating interface usability. Usability problems that can arise at interaction with the interface is the result of this method. In its essence, it is about how the user follows the denoted instructions. It uses theory about exploratory learning as presented in Section 2.2 and thus exploratory acting to evaluate the guessability of the interface, attempting to predict if the users will do right and why they might not do it.

Osvalder, Rose, and Karlsson (2015b) continues that the evaluators step through the

instructions in the HTA and for every sequence of actions the following four questions are asked.

1. Will the user try to achieve the correct effect?
2. Will the user discover that the correct action is available?
3. Will the user associate the correct action with the desired goal?
4. Does the user get feedback if the correct action is carried out?

Osvalder, Rose, and Karlsson (2015b) explains that each of the questions is either answered with a *yes* or *no*, together with a motivation for the answer. The reason and problem is also stated if the answer is no. A form for both CW and *predictive human error analysis* (PHEA) suggested by Osvalder, Rose, and Karlsson (2015b) is shown in Appendix B.

Finally, Osvalder, Rose, and Karlsson (2015b) describes that CW can be used at any time in the development process of a product, to finished product. Compared to user tests, it is inexpensive and resource-economic. This are the same characteristics as for *predictive human error analysis* (PHEA) seen in Section 3.3.4 with the following differences. It only analyzes how easy the interface is to learn, the guessability, and can be cumbersome and time-demanding for inexperienced evaluators, especially if many steps is exist. The interpretation of the interface by both new and experienced users are captured with CW.

3.3.4 Predictive Human Error Analysis

In order to get a deeper understanding of the errors produced by the CW as described in the previous section, they need to be analyzed. Brought up in its entirety by Osvalder, Rose, and Karlsson (2015b) and initially developed by Embrey (2004) is a method called *predictive human error analysis* (PHEA) that evaluates the interaction between an operator and technical system by finding potential usage errors as-well as their consequences. The result depends on the evaluators' sense of fantasy and creativity. PHEA tries to answer the what happens if the user does the following. The answers can then be used to identify actions in the HTA (Section 3.3.2) that are especially prone to produce usage errors.

1. Consequences of doing:
 - a. The correct action at the wrong time.
 - b. The incorrect action at the right time.
 - c. Nothing.

The analysis is carried out according to the following steps according to Osvalder, Rose,

and Karlsson (2015b).

1. Define the task in steps using the HTA.
2. In every step, identify possible action errors, and for each of them find out:
 - i. The reason of the error.
 - ii. The consequences.
 - iii. If the user will discover the error.
 - iv. If the user can recover from the error or correct it, and how.

Finally Osvalder, Rose, and Karlsson (2015b) explains that PHEA shares the same characteristics described in the end of Section 3.3.3, with the addition that it does not consider the underlying cognitive processes that leads to these errors occurring. A form for both CW and PHEA suggested by Osvalder, Rose, and Karlsson (2015b) is shown in Appendix B.

3.4 Interaction analysis by analytically evaluating the UI with CW and PHEA

In order to evaluate the interaction of the UI produced by Adelbäck and Malmgren (2015), the methodology proposed in Osvalder, Rose, and Karlsson (2015b) has been used for the rest of this section and is further described here. A flowchart for the process here can be seen in Appendix C.

3.4.1 Definition of the evaluation

The first step in the method proposed by Osvalder, Rose, and Karlsson (2015b) is to define the analysis by answering following five questions.

1. What is the purpose of the analysis?
2. What product is analyzed?
3. What usage is analyzed?
4. Who is the user?
5. What is the environment does the interaction take place?

The purpose of the analysis is already described in the introduction. The UI described in Adelbäck and Malmgren (2015) was the subject of analysis. The users were the authors of this report. The environment were an office environment in group rooms with little to no disturbances of the work. The UI was edited in Excel and the coding was done

in a text editor. Both of which were in the environment of a Windows computer. The human-technology system will be described in more detail in the next section.

3.4.2 Describing the human-technology system

In this step, a more thorough analysis of the user, environment, task and the UI was carried out. Osvalder, Rose, and Karlsson (2015b) propose using HTA (see Section 3.3.2) to further explain the task, sub-tasks and how they connect and interact with each other. The environment, users, and target group considered for the analysis should also be further detailed with focus on what aspects affect the interaction between the users and the UI. Finally, the UI itself should be explained in terms of how the UI works, layout and how it changes during interaction.

For this evaluation, the relevant aspects to go deeper into was the user, task and UI, leaving out the environment as it is described in the previous section.

The users were production engineering students in the end of their university education, with a past in an automation and mechatronics bachelor that have an interest in programming of different orientations, and have chosen all available courses regarding production simulation and AutoMod. They are new to the UI. General programming and usage knowledge about Excel exists from the university, without having any deeper knowledge in for example macros.

The tasks to be evaluated were the process of creating input as-well as output functionality from the UI to AutoMod from the perspective of a programmer. The input process is described in Appendix D, and the output process in Appendix E. They are based on spoken information from our ÅF supervisor Leo Adelbäck on how to program the UI.

3.4.3 Predictive Human Error Analysis

In the third step in the proposed methodology by Osvalder, Rose, and Karlsson (2015b), the PHEA methodology described in Section 3.3.4 was conducted given the groundwork done earlier in the previous sections.

In this project, it is worth clarifying that the interaction analysis methods were conducted both by the user and by a knowledgeable person in cognitive and physical ergonomics, interface design, and partly in the human-technology system being analyzed. This because at least one of the authors fulfills both of these criteria.

Thus, the problems displayed in the PHEA are the ones occurring in the programming of the UI for this project. As the CW is supposed to be based on the HTAs to generate errors, which are processes that have to be done in this project for the simulation model anyway, the CW of the HTAs was not conducted directly. Rather, the questions brought up in Appendix B for CW was used as inspiration when attempting to come up with errors that had to do with the HTAs.

3.4.4 Conclusion and follow-up

The final step in Osvalder, Rose, and Karlsson (2015b) proposed methodology is to conclude the findings from the CW and the PHEA and to investigate which of them that can actually occur in reality. The problems that have a chance of occurring in reality should then be the base for improvements and conclusions. There is always a chance of errors quickly discovered and adjusted, or that a way of completing the task is possible despite the problems.

4

Results

For the simulation model and then the UI evaluation, the results are presented in the two sections in this chapter.

4.1 Simulation model

The finished model is simplified in Appendix J. It consists of an operator, that can reject pallets which the service truck transports away. Accepted pallets go into the pallet breakdown cell that puts the boxes on the conveyor. These either go to the cranes, or past the cranes to the other operators if the boxes are erroneous, upon which they are corrected and sent to the cranes again. This is one of the flows.

The other flow is from the cranes, where a box can either go to rack, or to the operators at the manual handling stations (MHSs). The result of the operations include the box leaving the system, the original box being sent back to the cranes and a new box leaving the system, and the original box being sent back to the cranes and a new box being sent to racks.

There are two rack cranes that handles a number of racks each, and a rack truck that transports filled racks out of the system. Between all of the stations are conveyors with the same behavior as in the real system. There is a repair operator that repairs the machines or conveyors when they break down. Finally, there is a service truck that handles pallet buffers at the pallet breakdown cell, material to and from the operators, and some other miscellaneous tasks.

Worth mentioning is that, among other details, pallet covers and boxes leaving the MHSs are transported by the service truck and strapped by operator 1.

Screenshots of the result is shown in Figure 4.1 and 4.2.

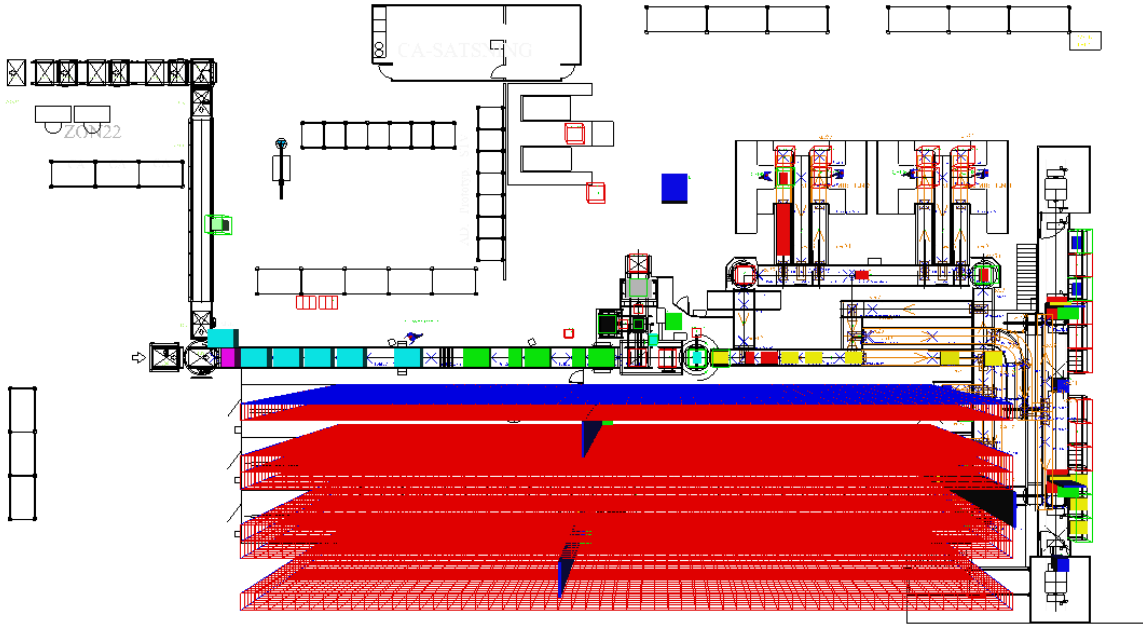


Figure 4.1: Screenshot overview of the simulation model created in AutoMod.

The simulation results are shown in Appendix K. Day-shifts for 5 days is the simulated time-span with 20 replications. In this scenario, the first resource never starves, and the cranes always have orders to send out from the storage. The simulated maximum capacity for boxes in to the system is 4046, 4950 for boxes leaving the cranes, out of which it is 2900 going out to racks (including boxes from the MHSs to racks).

The machines of interest and with the highest utilization is also presented in Appendix K. The bottleneck is revealed to be the cranes with 93%, 93% and 90% respectively (excluding downtime).

Following the current-state results (the first table) are the two future-state investigations in Appendix K. Here, a reject route described in Figure 4.3 is analyzed by request from the case company and is supposed to ensure that the bottlenecking cranes are always supplied with boxes in accordance with the theory of constraints described in Section 2.1. This by redirecting empty boxes and rejects via this new conveyor. One can see that the results are not improving by this change, but rather, they are declining.

The other future-state investigation consists of the eventual addition of a fourth crane. Due to time-limitations and implementation complexity, this was instead simulated by increasing the speed of all cranes by 33% as-well as decreasing the loading and unloading times by 33%. The total boxes leaving the system is increased to 6007, with an increase in boxes leaving via racks to 3452, and incoming boxes remaining the same. Here, the utilization of the cranes are naturally decreased as they are the bottleneck. Note that

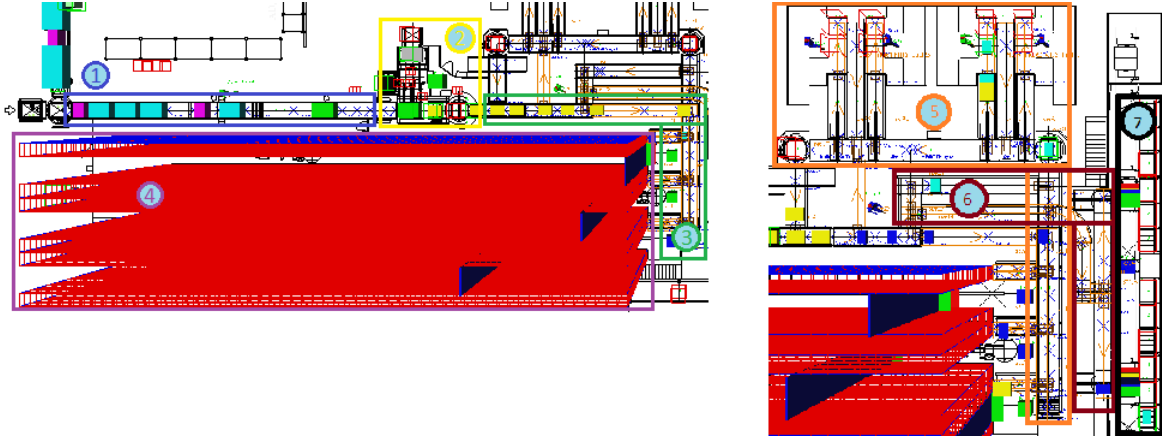


Figure 4.2: Screenshot of the different parts of the simulation model. Part 1 is the incoming pallets with boxes and operator 1. Part 2 shows the pallet breakdown cell. The incoming boxes enter the cranes in Part 3. Part 4 displays the three cranes with two sides each. The outgoing boxes from the cranes and the MHSs can be seen in Part 5, and an exit for the boxes going to the racks are shown in Part 6. Finally, the rack cranes, racks and rack truck are found in Part 7.

the utilization for the rack truck has now increased significantly to 72%.

Lastly, worth mentioning is the number of boxes coming in to the MHSs and being sent back to the cranes. Because of how the system is built up, the boxes to MHS are the boxes leaving the cranes subtracted by the boxes leaving the racks. Additionally, out of the boxes going in to the MHSs, the current scenario yields that it is about 75% of them going back to cranes. This means that the total number of boxes in to the cranes for the current-state is about $4064 + (4950 - 2900) \cdot 0.75 = 5601$. For the future-state with the decrease in crane C/T with 33%, about $4132 + (6007 - 3452) \cdot 0.75 = 6048$.

4.2 UI evaluation through PHEA

Once the simulation model was finished, the evaluation of the errors found during the model's creation could be analyzed. As most of the errors were already found by creating the model, there was no need to conduct a CW. Thus, the errors from the input process (Appendix D) and output process (Appendix E) are evaluated with PHEA in the following sections.

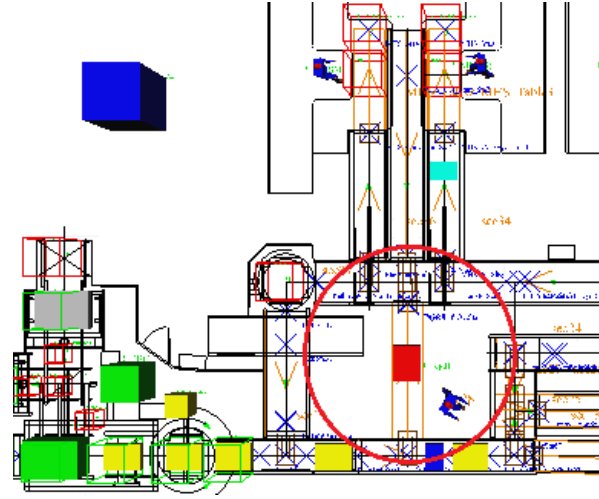


Figure 4.3: Separate conveyor for rejected and empty boxes evaluated in the future-state.

4.2.1 Input process

When creating many tables, Step 1.3 in Appendix D takes a long time and errors occur as it demands repeatable memorization of start and end cell for each table. Errors occur at compilation if there are too few or too many variables specified for the table that has been read to a file. The reason is that the UI is not designed to aid the user in this aspect. It is recoverable by bug-fixing and interpreting compilation errors.

The same argument as above can be made for Step 1.2. It is a lot of manual work transferring the data from the model specification to the UI if everything that can be changeable should be changeable, which in practice means all input data to the model. The errors occurring are mismatches between description and value in different ways. For example, two times being switched with each other, unclear descriptions leading to the wrong time being used, or that the UI and the model have the variables inserted in different orders. The consequence is that the model will not pass verification and the error is discovered when running and inspecting the model. Of course, there is a risk that the error never will be discovered, or after a long time after the simulation model is complete. If the detailed and sometimes complex data is not gone through carefully before running the model, it has to be done at a later stage when the model is run and the input variables are found to be erroneous. This careful comparison of model specification and UI variables is the recovery of the error.

Additionally, the same amount of manual work and comparison as previously described is the reason for errors when it comes to Step 1.4. Here, the coded variable name should correspond to that of the UI to avoid using the wrong time for processes. It is also

discovered at the running of the model and during verification. Recovery is possible after finding the bug and changing the name.

Step 1.4 takes a long time to complete as there was no naming conventions available and they had to be invented by the authors. This does not quiet fit in the PHEA, but it is important to bring up anyway.

Step 1.5 produced many compilation errors as the authors previously had only defined variables in AutoMod, not in the file (.asy) that AutoMod produces from its interface. Trial and error is the reason for these errors as there does not exist any written instructions or manual. The consequence is extra time being spent bug-fixing the compilation errors. Discovery can be done on the first compilation, but also later in the project when the variables are put into use and they turned out to have the wrong dimensions resulting in logical errors, and so on.

An error occurred when following Step 1.1 and constructing the tables vertically instead of horizontally. If an additional variable has to be introduced, the tables following the changed one has to be moved if there is not enough space. This means that the tables have to be moved again as part of Step 1.1, that Step 1.3.1 and 1.3.2 has to be done again for all the affected tables, and finally that the variables have to be added to the code again as seen in Step 1.5. The reason for this is the same as in the previous paragraph, namely the lack of a manual for the programmers.

4.2.2 Output process

When designing an output sheet for the utilization of the machines in the model (Step 2.4 in Appendix E), the wrong cell numbers were accidentally entered. This meant that all the prepared cells for min, average and max (end of Step 2.1 and Step 2.2) was overwritten with the utilization data. Since this action was irreversible, these two steps had to be done again as a consequence.

To clarify, the reason was that the data for two adjacent rows were interchanged. The reason for this, in turn, could possibly relate to that the sheet names and text names were not easy enough to distinguish from each other. Thus, it is believed it was a human error. Discovery was done immediately after reading the outputs from the files. The correction that had to be done was to eventually switch back the data on the two rows.

Step 2.3 and Step 2.4 lack steps that has to do with ensuring that the Excel-sheet in the UI and the text-file generated by AutoMod follows the same order of the entities. It is easy to make errors that are hard to discover, as for example all resources have

the same categories of data and the same length. This results in that the error is not discoverable without already knowing the outcome, more or less, and might even be undiscoverable if the data sets are very similar to each other. The consequence is that the UI to an extent presents the wrong data. The reason for the error occurring is that the HTA, and therefore the instructions available are not including, emphasizing or considering this error nor its severity. It is corrected by manually controlling the order of the output text-file with the information in AutoMod, and that the UI follows the same order.

An error occurred at simulation when generating the output-files with the message *Error number 1004: Application-defined or object-defined error*. The discovery is upon run, after one or more changes has been done to the UI. The message presents no indication of the reason for the error, and it was discovered because small changes was done between every simulation, resulting in the user knowing immediately what the error was. In other cases, the reason would be unknown, and the correction being searching for the bug. As it turned out, the real error was because no cells had been inserted in the control sheet for an output-file as a part of Step 2.4. The correction is filling in the correct cell data.

5

Discussion

Below, the method followed by the result is discussed for both the simulation model and the evaluation. Finally, a discussion about sustainability aspects are presented in the last section.

5.1 Method

CW and PHEA along supporting methodology provided by Osvalder, Rose, and Karlsson (2015b) are generic methods to evaluate the usability of products of different complexity as seen in Sections 3.3.3 and 3.3.4. Our opinion is that they have fulfilled the purpose of this report seen in Section 1.2 by evaluating the usability of the UI via the HTAs and the methods, and having generated the errors and clearly described their reasons, consequences, discoverability and recoverability.

In our project, the errors were mainly generated from programming the UI and the CW methodology was not used. A reason for this is the lack of written programming instructions which in turn lowers the quality of the HTAs because they are based on spoken instructions. Another reason include time limitations as a majority of the time had to be spent on creating the model. Because one of the authors have experience in these methods, it would further secure that the project was completed on time. Since the purpose of CW is to generate errors, and that the alternative is testing on real users under surveillance by the evaluators, the purpose of CW is already fulfilled by the authors being both users and evaluators. There might however be undiscovered errors because of what the HTAs are based upon, as previously described.

However, there exist methods recommended for evaluating simulation UIs specifically, as described in Montgomery (2011). It was not used due to time limitations as it was discovered relatively late in the project, as-well as not being accessible through the university library and inducing unexpected costs. This assumedly has a better effect

than our generic methods, but as previously said, this methodology is still valid for the chosen human-technology system and fulfills the purpose.

5.2 Result

The result of simulation model and the evaluation are both discussed below.

5.2.1 Simulation model

We were surprised that the current-state results resulted in such low standard-deviations. This is positive since it means that the model is more trustworthy as one of the validation steps in Section 3.1.2 is about making sure that the results of the model are not stochastic. It also means that the maximum capacities found can be trusted with higher certainty.

Introducing a shorter route for the rejected and empty boxes having no effect was also very surprising to us, as-well as the case company. The only effects it had on the system were negative ones, as seen in the minimum values for both capacities and utilization factors. This has to do with the fact that implementing this route was harder than anticipated, and introduced too many bugs to the system. The bugs meant that there would be stoppages in the conveyor system, and could not be addressed completely due to time reasons. However, one can see from the maximum values that they even in the best cases are not an improvement to the system, and thus a wasteful investment. Worth noting is that despite the lower minimum values compared to the previous runs, the average values are only changed slightly, indicating a rare bug.

The only future-state proposal by the case company that upon investigation gave positive results was the idea of introducing another crane. Theoretically, one can simply decrease the cycle-time (C/T) by 33% when going from three to four cranes, which was done due to time and implementation complexity reasons. The only parts not captured by doing this is the additional buffer space, storage space, and conveyor changes this would induce. Although, the latter parts are considered of lesser importance when compared to the purpose of introducing the fourth crane, namely to bring down the utilization by decreasing the aggregated C/T .

The effect was not as big as anticipated. The reason is surprising but understandable; that the combination of low conveyor speeds, largely different spacing on the incoming conveyor resulting in a unnecessarily high C/T of the conveyor, and pallet breakdown

cell characteristics (such as the cell stopping when one of the two pallet buffers become full), are now not enough to supply the cranes with more boxes. This can be seen by the very slight increase in boxes coming in to the system.

The lower utilization of the cranes also has to do with the counters existing to prevent them from flooding the system with boxes when there are many orders. This means that the resources following the cranes most likely have to be increased in their footsteps. Namely, the rack cranes, rack truck, MHSs, and conveyors are candidates to this.

In order to relate the utilization of the cranes with the maximum capacities, one can compare the total boxes handled by the cranes before and after the improvements. The current-state has about $5601 + 4950 = 10551$ total boxes handled, while the future-state with improved crane C/T has $6048 + 6007 = 12055$. An increase by 14% does not seem much, but given that the utilization of the cranes have dropped to about 78%, that there are returning boxes not displayed in this result, and arguments made in the previous paragraphs, it seems reasonable to us that there was not a higher benefit from increasing the cranes' C/Ts by 33%.

Looking back at the introduction in Section 1, the objective of finding out the maximum capacities has been fulfilled, and some future-state analysis has been done on the system.

5.2.2 Evaluation

The input interaction analysis brings up usability problems with the UI. This was an unexplored aspect of Adelbäck and Malmgren (2015), and the results confirm this.

For many of the errors, one can say that they depended at least partly on there not existing a programmer's manual, and that the information on how to program the UI was presented to us orally by our ÅF supervisor and creator of the UI Leo Adelbäck himself, which is what the HTAs are based upon. Thus, we suggest that a programmer's manual to be created to aid both new and experienced simulation programmers in learning how to program the UI and to decrease the time the creator of the UI has to spend on supporting it.

Additionally, there is a need to aid the user in making sure that the UI and AutoMod sides match without inducing unnecessary cognitive pressure. This will result in fewer errors are being made, and that the UI programming can be more effective. It could also be corrected by otherwise ensuring that a previously specified order is followed both when constructing the tables in the UI and coding the variables in AutoMod. An even better solution however is if one could include variable type, name, capacities, dimensions, etc. in the UI and eliminate the Steps 1.3 and 1.4 entirely. The same

argument can be made for Steps 2.3 and 2.4.

In order to prevent mixing up the output data in Steps 2.3 and 2.4 between the UI and the code, the resource name could be printed with the output data. This will ensure that the correct data is assigned to the right resource in the UI.

Furthermore, the feedback given from the UI to make the user discover errors are questionable when analyzing the results. This because the errors produced not sufficiently aid the user in correcting the error. Therefore, this report recommends looking more into how the feedback is in the system, how it should be, and then do improvement work accordingly.

There is also a lack of completely describing and analyzing the human-technology system for the UI according to the suggested methodology in Bohgard et al. (2015), and is recommended to design the UI and address its problems in a thorough way. The parts from the methods of Bohgard et al. (2015) we recommend to be further explored for the primary user group (assumedly experienced programmers at ÅF) are data collection, system and task analysis, more detailed user description, and finally heuristic pressure analysis. Additionally, we recommend that the primary user group is evaluated with the methods used in this report as-well. More iterations of evaluating has to be done when the improvements to the errors have been implemented, in order to see that they had the desired effect.

We were surprised that the interaction analysis produced this many errors for us - the users in the evaluation. Going into the project, our feeling was that the interaction analysis would get thin and lack results, despite the fact that this was at least not directly explored in Adelbäck and Malmgren (2015).

This fulfills ÅF's interest in improving their methods as seen in Section 1, as we have pointed out errors and helped them with improvement suggestions as-well as future research suggestions.

5.3 Sustainability

The project was conducted with the underlying assumptions of sustainability, but efforts to actively involve it in the overall work was not done. However, sustainability factors can still be motivated and discussed for the work done in this project.

Economical sustainability is easily motivated for this project. A tool for decision-making, to for example be more efficient in their work and reduce waste, can be trans-

lated into economical benefits such as reduced costs for stress-related injuries and the worker being more effective for his salary. Making the programming of the UI more effective also has positive effects for ÅF when it comes to lowering costs for them, and potentially for the customer, while maintaining the same quality or even improving it.

On a higher level, the simulation is used to shift the time from building and testing in real life, to simulating in the computer. The cost of the simulation model is often low compared to the cost, time and quality of testing in a real system. This is especially true for when constructing a completely new system, and having no idea if it will work efficiently can be an immensely costly affair. Therefore, it has a payback time that can be calculated from the reduced costs as a result of using simulation. This is the primary reason as to why simulation is economically sustainable.

The social sustainability is connected to the efficiency points made in the previous paragraph. A worker will benefit from doing the right things at work, such as not inducing or having to work with waste like overproduction. However, in a western company with top-down management, this is far from easy as the task of improving the work lies primarily on the management and not the workers (Liker and Hoseus 2008). This means that a western company attempting to become lean (like the case company) without changing their culture fundamentally to bottom-up, will inevitably have workers that are unmotivated to do improvement work. Temporary gains from the *tools* of lean will be a reality - actually becoming lean in the long run will not, as western companies usually just cherry-pick the easiest and believed most beneficial tools to implement.

The workers can benefit greatly from the management investing time and effort to keep the working conditions up-to-date. By addressing problems that the simulation model confirms, the workers feel that the management does something about the problems. It can be improving how they work to better suit the system and to even out imbalances, reducing waste of different kinds, and so on.

For this project, the effects on the environmental sustainability aspects are generally connected to the shift from testing to simulation as discussed earlier. This because avoiding to buy unnecessary industry equipment has a lower impact on the environment, as it is often a matter of products made of metal that has been shipped around the world, and so on.

5.4 Contribution to current research

As the background in the introduction states, this project aimed to contribute to current research by continuing the development of a production DES UI by evaluating its usability. This for new users recently educated in programming and in AutoMod at university-level. This is motivated for a couple of reasons. The first being the lack of evaluating the UI by and from Adelbäck and Malmgren (2015). Since it is a recently produced UI that assumedly contributes to current research as a master's thesis, improving the UI is considered to be contributing to current research as-well. Secondly, and lastly, is that how usability of the UI has a very large impact on the simulation quality for the customer according to Montgomery (2011). This shows the importance of usability for the customer, and can be motivated for the programming of the UI too, as motivated in the end of the background.

The theory used in Adelbäck and Malmgren (2015) does only include what Bohgard et al. (2015) calls data collection, namely interviews, to base improvement potentials for the UI. This was mainly done from the perspective of the customer and not of the programmers. The theory and methods in our report are part of what is needed to get a more complete picture on how the result actually turned out, namely interaction analysis and evaluation which is of benefit not only for the programmers of the UI as conducted in this report, but also for the customer using the UI if decided to be carried out for them as-well. Our theory focuses on the interaction from a objective point of view, as one of the authors are experienced in the methods displayed in Bohgard et al. (2015). This as opposed to Adelbäck and Malmgren (2015), which has a subjective and qualitative approach, where the opinions of the programmers play a big part. Optimally, one can do both and get a more complete picture.

This report therefore contributes to the current research of improving implementation times of a production DES UI.

6

Conclusion

As the introduction entails, this project was about evaluating the usability of newly created UI, as the UI has a very large effect on simulation quality and usability (Montgomery 2011). The theory and methodology are generic for the purposes of simulation model-creation and usability evaluation.

A result was the complete simulation model of the current-state according to the specifications of the case company, including programming the UI to make it accessible to them. This results in a decision-making tool for the company. The model was conducted from the ground up, starting with data collection.

The main maximum capacities found were boxes in to the system (4064), boxes leaving the cranes (4950) and boxes out from racks (2900) for a 5 day period with 9 hour shifts including breaks.

The cranes were found to be the bottleneck with close to 100% utilization. A shorter and less wasteful route for boxes was tested, but resulted in no improvement compared to the current-state. Additionally, the storage cranes C/Ts were increased by 33% to simulate the aggregated effect of putting in an additional crane. Not surprisingly, this gave the most effect on maximum capacities as the cranes were the bottleneck. Thus, increasing the maximum capacities is done by eliminating the bottleneck that is the storage cranes.

The evaluation of the UI resulted in the recommendations of creating a programmer's manual, help the user to match the UI and the AutoMod environments to each other, improving error messages given by the UI, and properly describing the human-technology for the intended users. New iterations of the evaluation has to be done when improvements have been implemented to prevent the errors, to see if they have the desired effect.

The results in this project are the type of results expected from using the methodology of simulation according to Banks (1998a), of bottleneck analysis by utilization as seen in

Lima, Chwif, and Barreto (2008), and the usability evaluation in the form of interaction analysis brought up in Bohgard et al. (2015).

Noteworthy limitations of the project include time and resources, where the primary focus has been on quality of the simulation model for the case company.

This project contributes to current research by evaluating a newly created UI (2015) for production DES and AutoMod, and thus continuing the work of the UI.

Given what is displayed above, the research questions have been answered.

Thus, future research include the following.

- Constructing a manual for the UI.
- New iterations of interaction analysis for the work instructions from this manual.
- Update the UI to improve its usability based on the result of this report and those of future iterations.
- Look into extending the evaluation of the UI to include more detailed aspects other than interaction analysis, namely data collection, system and task analysis, more detailed user description, and pressure analysis, as-well as do all of these for the primary user group.
- Possibly use better and recommended evaluation methods like Montgomery (2011).
- Construct code for extracting start and end of active times for resources into text-files in order to use the active period method.
- Balancing current-state and future-state.
- Further implement changes of the system when deemed necessary.

Bibliography

- Adelbäck, L. and R. Malmgren (2015). *Generic User Interface for Simulation*. (Accessed: 3 May 2018). Göteborg: Department of Product and Production Development, Production Systems, Chalmers University of Technology. URL: <http://publications.lib.chalmers.se/records/fulltext/220568/220568.pdf>.
- ÅF, ÅF Industry AB. (n.d.). “AutoMod”. In: *AutoMod - Simulering av produktion och logistik - Programvara*. (Accessed: 25 May 2018). URL: <http://www.automod.se/automod.html>.
- Banks, J. (1998a). “1.7 Steps in a Simulation Study”. In: *Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice*. Hoboken, New Jersey, USA: John Wiley & Sons. DOI: 10.1002/9780470172445.
- (1998b). “22. Guidelines for Success”. In: *Handbook of Simulation - Principles, Methodology, Advances, Applications, and Practice*. Hoboken, New Jersey, USA: John Wiley & Sons. DOI: 10.1002/9780470172445.
- Bohgard, M. et al. (2015). 3rd ed. Stockholm: Prevent. ISBN: 978-91-7365-195-0.
- Chalmers, Chalmers University of Technology. (2016). “MPR271 - Simulation of production systems”. In: *Chalmers: The Student Portal*. (Accessed: 15 February 2018). URL: https://student.portal.chalmers.se/en/chalmersstudies/courseinformation/Pages/SearchCourse.aspx?course_id=25805&parsergrp=3.
- Danielsson, M. (2001). *Teknisk psykologi*. Stockholm: Natur och Kultur. ISBN: 978-91-2770-660-6.
- Dawson, C. and H. Lindahl (2017). *Production Flow Simulation Modelling in the Foundry Industry*. (Accessed: 11 April 2018). Göteborg: Department of Product and Production Development, Production Systems, Chalmers University of Technology. URL: <http://publications.lib.chalmers.se/records/fulltext/250247/250247.pdf>.
- diSessa, A. A. et al. (1995). “Computers and Exploratory Learning: Setting the Scene”. In: *Computers and Exploratory Learning*. Ed. by A. A. diSessa et al. Berlin, Heidelberg: Springer, pp. 1–12. DOI: 10.1007/978-3-642-57799-4_1.

- Embrey, D. (2004). “Qualitative and Quantitative Evaluation of Human Error in Risk Assessment”. In: *Human Factors for Engineers*. Ed. by C. Sandom and R. S. Harvey. London, UK: The Institution of Engineering and Technology, pp. 151–202. ISBN: 978-0-86341-329-2.
- Eriksson, U. (2005). “Diffusion of discrete event simulation in Swedish industry on the way to an increased understanding”. In: *Doktorsavhandlingar vid Chalmers Tekniska Högskola*. Göteborg: Industrial and Materials Science, Chalmers University of Technology. ISBN: 91-7291-577-3.
- Goldratt, E. M. and J. Cox (2004). 3rd rev. ed. Great Barrington, Massachusetts: North River Press. ISBN: 0-88427-178-1.
- Goodwin, A. and D. Pantzar (2017). *Improving a production flow using Value Stream Mapping and Discrete Event Simulation*. (Accessed: 11 April 2018). Göteborg: Department of Product and Production Development, Production Systems, Chalmers University of Technology. URL: <http://publications.lib.chalmers.se/records/fulltext/250248/250248.pdf>.
- Hollocks, B. W. (2006). “Forty Years of Discrete-Event Simulation: A Personal Reflection”. In: *The Journal of the Operational Research Society*. Vol. 57. 12. London, UK: Palgrave Macmillan Journals, pp. 1383–1399. URL: <http://www.jstor.org/stable/4622639>.
- Janhager, J. (2005). “User Consideration in Early Stages of Product Development : Theories and Methods”. QC 20100618. PhD thesis. KTH, Machine Design (Div.)
- Lewis, C. and C. Wharton (1997). “Chapter 30 - Cognitive Walkthroughs”. In: *Handbook of Human-Computer Interaction (Second Edition)*. Ed. by M. G. Helander, T. K. Landauer, and P. V. Prabhu. 2nd ed. Amsterdam: North-Holland, pp. 717–732. ISBN: 978-0-444-81862-1. DOI: doi.org/10.1016/B978-044481862-1.50096-0.
- Liker, J. K. and M. Hoseus (2008). *Toyota culture: The heart and soul of the Toyota way*. New York: McGraw-Hill. ISBN: 978-0-07-149217-1.
- Lima, E., L. Chwif, and M. R. P. Barreto (2008). “Metodology for Selecting the Best Suitable Bottleneck Detection Method”. In: *2008 Winter Simulation Conference (WSC)*. Miami, Florida: Winter Simulation Conference, pp. 1746–1751. ISBN: 978-1-4244-2708-6. DOI: [10.1109/WSC.2008.4736262](https://doi.org/10.1109/WSC.2008.4736262).
- Mansour, M. O. and M. Moreira (2017). *Improving Car Dismantling via Discrete Event Simulation*. (Accessed: 11 April 2018). Göteborg: Department of Product and Production Development, Production Systems, Chalmers University of Technology. URL: <http://publications.lib.chalmers.se/records/fulltext/251787/251787.pdf>.
- Montgomery, B. (2011). *The Impact of the User Interface on Simulation Usability and Solution Quality*. Ann Arbor, USA: ProQuest Dissertations Publishing. ISBN: 978-11-2478-286-7. URL: <https://search.proquest.com/docview/884331131>.
- Osvaiger, A.-L., L. Rose, and S. Karlsson (2015a). “9.3.1 Användarprofiler”. In: *Arbete och Teknik på Människans Villkor*. Ed. by M. Bohgard et al. 3rd ed. Stockholm: Prentice, pp. 494–496. ISBN: 978-91-7365-195-0.

- Oswalder, A.-L., L. Rose, and S. Karlsson (2015b). “9.6 Interaktionsanalys”. In: *Arbete och Teknik på Människans Villkor*. Ed. by M. Bohgard et al. 3rd ed. Stockholm: Prevent, pp. 512–519. ISBN: 978-91-7365-195-0.
- Oswalder, A.-L. and P. Ulfvengren (2015). “7. Människa-tekniksystem”. In: *Arbete och Teknik på Människans Villkor*. Ed. by M. Bohgard et al. 3rd ed. Stockholm: Prevent, pp. 355–357, 433–434. ISBN: 978-91-7365-195-0.
- Roser, C., M. Nakano, and M. Tanaka (2002). “Shifting bottleneck detection”. In: *2002 Winter Simulation Conference (WSC)*, pp. 1079–1086. ISBN: 0-7803-7614-5. DOI: 10.1109/WSC.2002.1166360.
- Roser, C. and M. Nakano (2015). “A Quantitative Comparison of Bottleneck Detection Methods in Manufacturing Systems with Particular Consideration for Shifting Bottlenecks”. In: *Advances in Production Management Systems: Innovative Production Management Towards Sustainable Growth*. Cham, Switzerland: Springer International Publishing, pp. 273–281. ISBN: 978-3-319-22759-7. DOI: 10.1007/978-3-319-22759-7_32.
- Sargent, R. G. (2013). “Verification and validation of simulation models”. In: *Journal of Simulation* 7.1. DOI: 10.1057/jos.2012.20.
- SAS, SAS Institute Inc. (n.d.). “Core Capabilities of JMP - Basic Data Analysis”. In: *JMP - Statistical Discovery. From SAS*. (Accessed: 19 March 2018). URL: https://www.jmp.com/en_us/software/data-analysis-software/core-capabilities-of-jmp.html#Basic-Data-Analysis.
- Stanton, N. A. (2006). “Hierarchical task analysis: Developments, applications, and extensions”. In: *Applied Ergonomics* 37.1. (Special Issue: Fundamental Reviews), pp. 55–79. ISSN: 0003-6870. DOI: <https://doi.org/10.1016/j.apergo.2005.06.003>.

Appendices

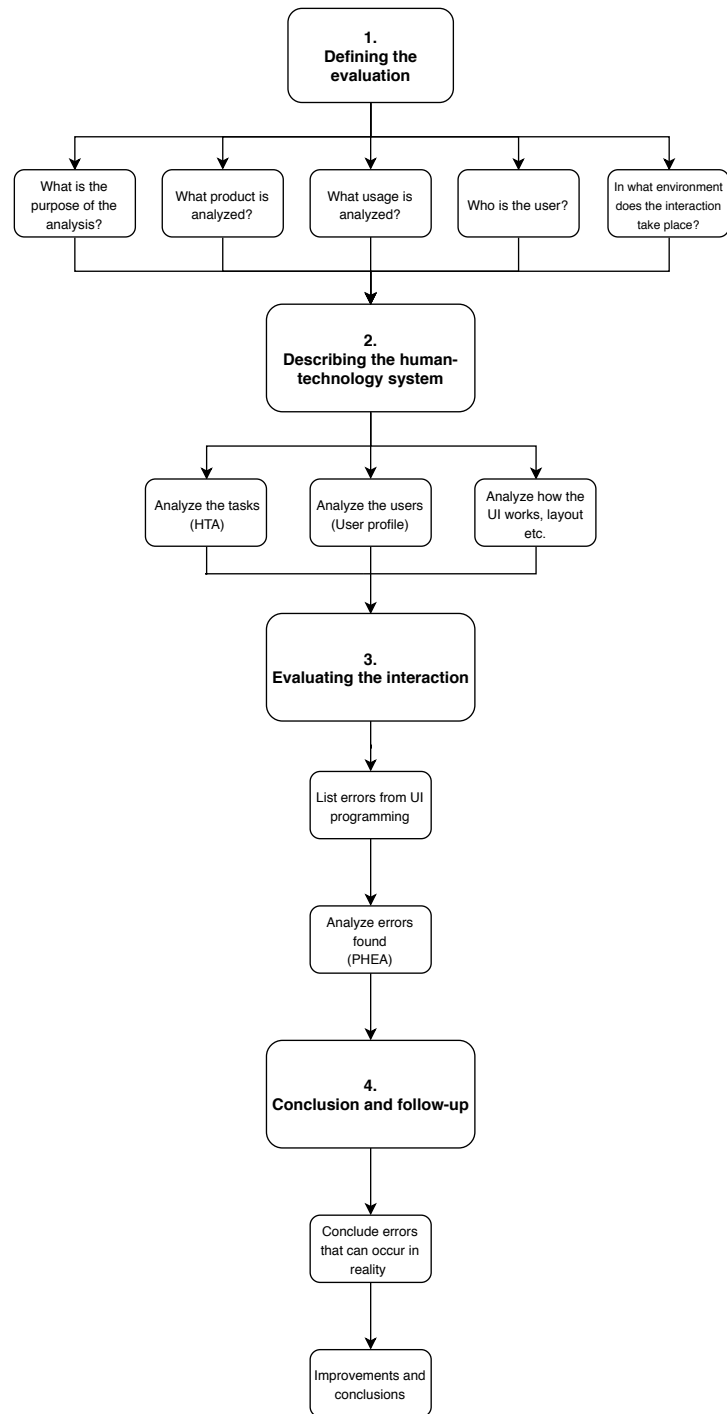
A Stakeholder analysis

	Primary	Secondary
Result	Volvo Trucks: Simulation model, bottleneck analysis, improvement suggestions, documentation, manual. ÅF: Image, responsibility towards customer. Chalmers: Deliverables.	Volvo Trucks: Examine relations in the model. ÅF: Improve templates, standardizations, feedback on methods, etc.
Realization	ÅF: Image. Chalmers: Project reaching master-level.	ÅF: Evolve way of working in projects and research if it can be connected.

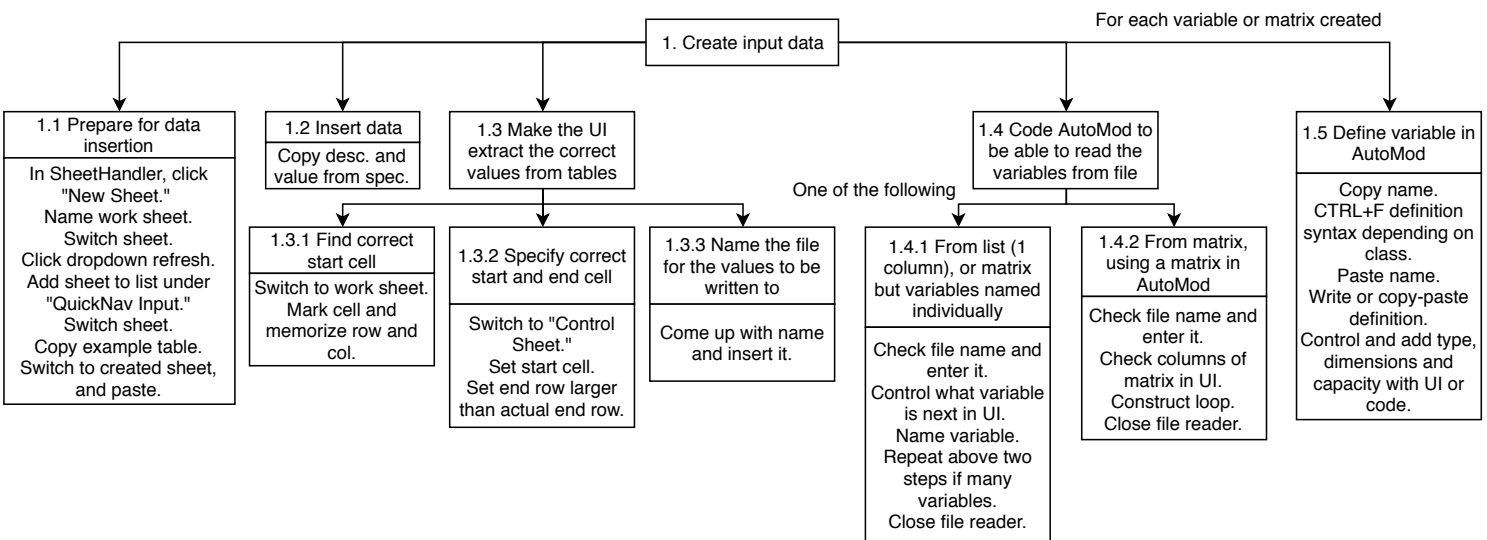
B
CW and PHEA form

Form for CW and PHEA (Osvalder, Rose, and Karlsson (2015)).					
CW			Yes/No	Why	Problem
1. Will the user try to achieve the right effect?					
2. Will the user notice that correct action is available?					
3. Will the user associate the correct action with the right effect?					
4. If correct action, will the user see that the action have brought the action have progressed the completion of the task?					
PHEA			Notes		
What action can the user do wrong at the right time?					
What action can the user do right at the wrong time?					
What happens if the user does a complete action or does nothing?					
What happens if the user carry out the actions in the wrong order?					
Error	Reason	Consequence	Discovery	Recovery/correction	

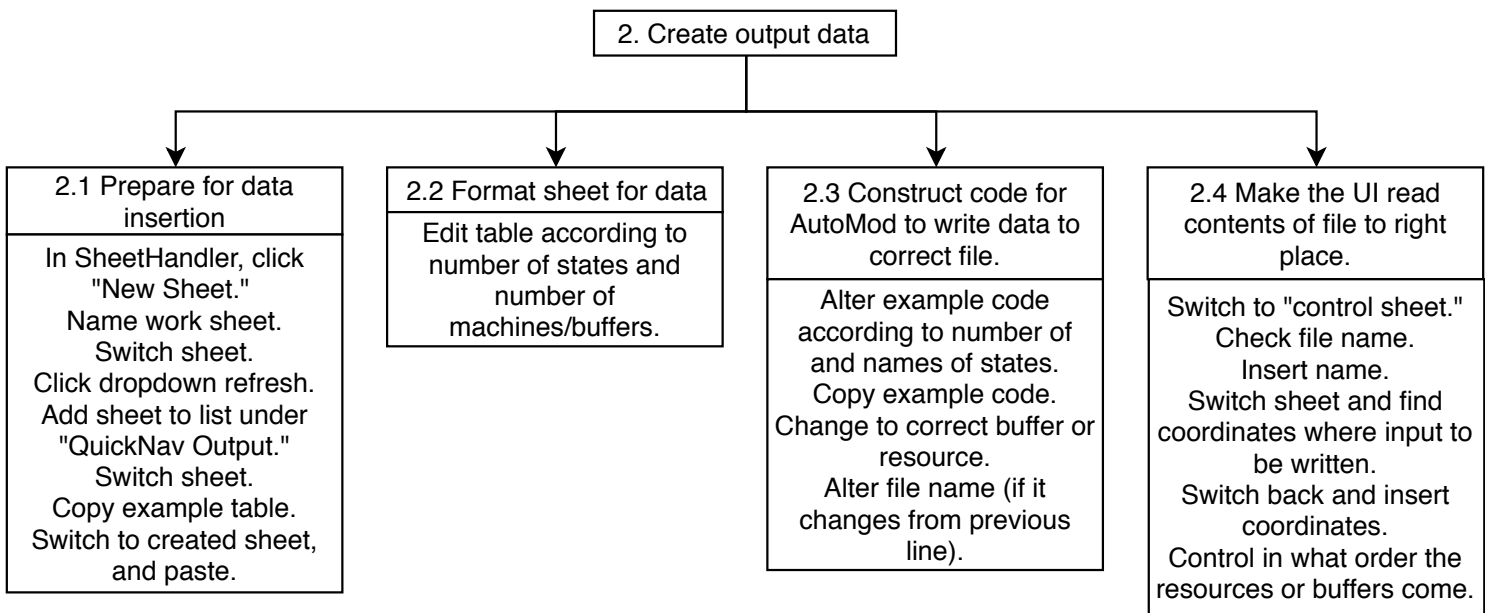
C Evaluation method



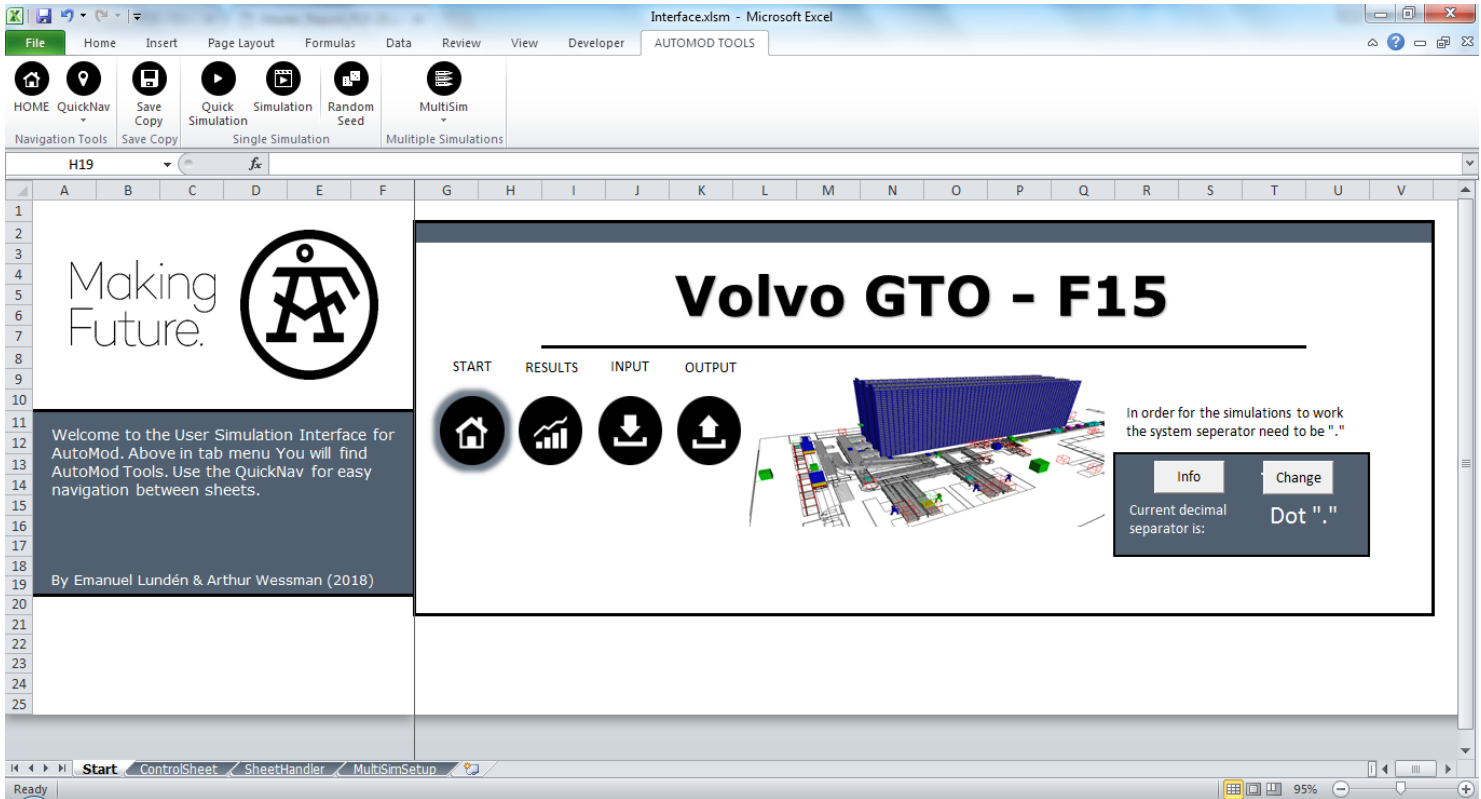
D HTA for the UI input process



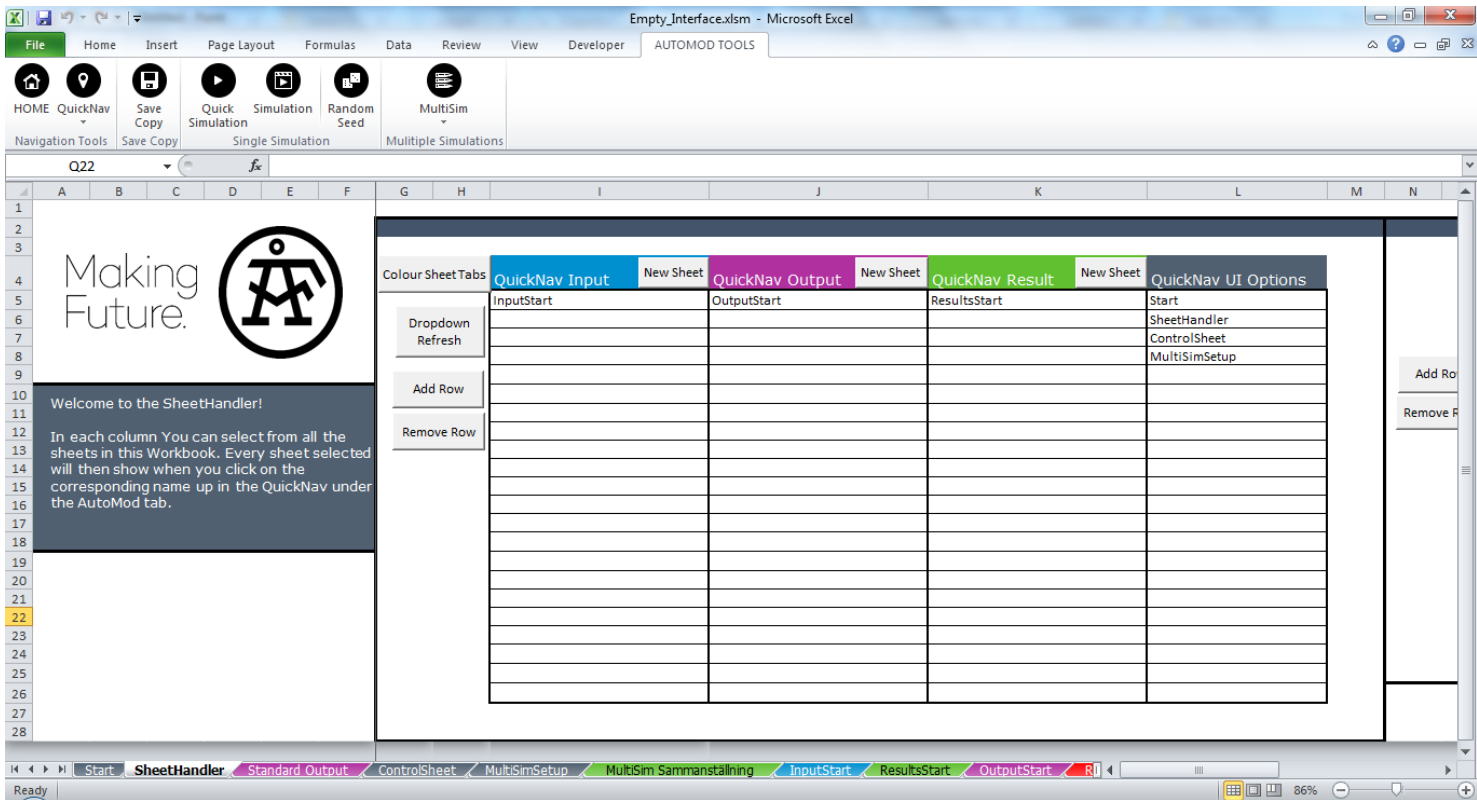
E HTA for the UI output process



F UI - Main page



G UI - Sheet handler



Making
Future.

Welcome to the MultiSimSetup!

Here You setup multiple simulation runs, and with repetitions of each run.

Analysis type

Vary factors

Repetitions

2

CRN

On

Factors

2

Response

3

Nr of Runs

1

Edit

Edit

Factor 1

Factor 2

1		
2		
3		

Response 1

Response 2

Response 3

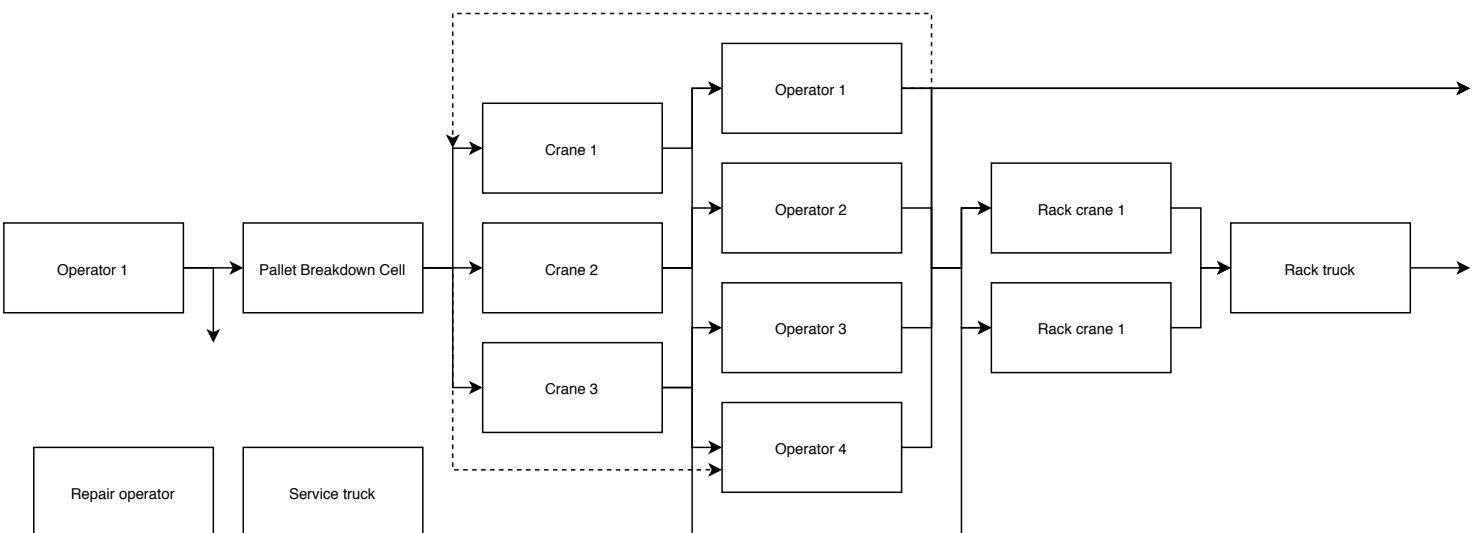
Edit

Edit

Edit

Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev

J Graphical model of the system simulated



K Simulation results

Boxes leaving the cranes				Boxes out from racks				Boxes in				Util crane 1				Util crane 2				Util crane 3				Util repair				Util Operator inbound				Util service truck				Util rack truck			
Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev
4705	4950	5130	126.8	2678	2900	3037	94.35	3839	4064	4277	102.1	0.892	0.93	0.948	0.014	0.894	0.93	0.948	0.014	0.851	0.902	0.949	0.028	0.061	0.106	0.154	0.029	0.504	0.536	0.567	0.014	0.245	0.257	0.269	0.006	0.483	0.542	0.603	0.026

With reject route active.

Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev
2954	4895	5203	471.4	1706	2844	3050	277.3	2334	3997	4287	400.1	0.535	0.917	0.954	0.091	0.534	0.915	0.953	0.091	0.527	0.886	0.943	0.088	0.075	0.116	0.212	0.034	0.307	0.53	0.568	0.054	0.149	0.255	0.273	0.026	0.317	0.538	0.6	0.057

With crane speed *1.33

Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev	Min	Avg	Max	StdDev
5739	6007	6161	119.1	3226	3452	3573	89.54	3941	4132	4328	91.25	0.763	0.789	0.816	0.012	0.761	0.784	0.808	0.011	0.718	0.768	0.794	0.019	0.055	0.1	0.144	0.023	0.567	0.582	0.614	0.011	0.284	0.291	0.302	0.005	0.688	0.719	0.745	0.014